Iterative receivers: scheduling, convergence speed and complexity

Salim Haddad

N° d'ordre : 2012telb0257

Sous le sceau de l'Université européenne de Bretagne

Télécom Bretagne

En habilitation conjointe avec l'Université de Bretagne-Sud

Ecole Doctorale – SICMA

Récepteurs itératifs : ordonnancement, convergence et complexité

Thèse de Doctorat

Mention : STIC (Sciences et Technologies de l'Information et de la Communication)

Présentée par **Salim HADDAD**

Département : Electronique

Laboratoire : Lab-STICC / Pôle CACS

Directeur de thèse : Michel Jézéquel

Soutenue le 16 novembre 2012

Jury :

M. Fabrice Monteiro,	Professeur à l'Université de Lorraine	(Président)
M. Maryline Hélard,	Professeure à l'INSA de Rennes	(Rapporteur)
M. Guido Masera,	Professeur à Politecnico di Torino	(Rapporteur)
M. Smail Niar,	Professeur à l'Université de Valenciennes	(Examinateur)
M. Philippe Coussy,	Maître de conférences/HDR, à l'Université de Bretagne-Sud	(Examinateur)
M. Amer Baghdadi,	Maître de conférences/HDR, à Télécom Bretagne	(Encadrant)
M. Michel Jézéquel,	Professeur à Télécom Bretagne	(Directeur)

Contents

Introduction **1**

1 Wireless Digital Communication Systems and Iterative Processing **5**

1.1 Wireless Digital Communication Systems 6

1.1.1 Convolutional Turbo Codes (CTC) 7

1.1.1.1 CTC Encoder 7

1.1.1.2 CTC Interleaver 10

1.1.1.3 CTC Puncturing 10

1.1.2 Bit-Interleaved Coded Modulation (BICM) 11

1.1.3 Mapping 12

1.1.3.1 Quadrature Amplitude Modulation (QAM) 12

1.1.3.2 Gray Mapping 13

1.1.3.3 Signal Space Diversity (SSD) 13

1.1.3.4 Constellation Sub-Partitioning Technique 14

1.1.3.5 Bits-to-Symbol Allocation Technique 15

1.1.4 MIMO Techniques 15

1.1.4.1 Space-Time Coding (STC) 15

1.1.4.2 Spatial Multiplexing (SM) 16

1.1.5 Channel Models 16

1.1.5.1 Frequency Selective Channel 16

1.1.5.2 Time Selective Channel 17

1.1.5.3 Double Selective Channel 18

1.2 Iterative (Turbo) Processing 18

1.2.1 Turbo Decoding 18

1.2.2 Turbo Demodulation with Turbo Decoding 18

1.2.3 Turbo Equalization with Turbo Decoding 18

1.2.4 Turbo Equalization with Turbo Demodulation and Turbo Decoding 19

1.3 Summary 20

2 Turbo Decoding: Algorithms and Schedulings 21
2.1 State of the Art . . . 22
2.2 SISO Decoding Algorithms . . . 22
2.2.1 MAP Decoding Algorithm . . . 23
2.2.2 Max-Log-MAP Decoding Algorithm . . . 24
2.2.3 Parallelism in Turbo Decoding . . . 25
2.2.3.1 Metric Level Parallelism . . . 25
2.2.3.2 SISO Decoder Level Parallelism . . . 26
2.2.3.3 Parallelism of Turbo Decoder . . . 27
2.3 SISO Turbo Decoding Schemes . . . 28
2.4 Turbo Decoding Schedulings . . . 29
2.4.1 Classical Turbo Decoding Schedulings . . . 29
2.4.2 Shuffled Turbo Decoding scheduling with Overlapping . . . 30
2.4.2.1 *Overlapping 1* scheme . . . 31
2.4.2.2 *Overlapping 2* scheme . . . 33
2.5 Summary . . . 34

3 Optimized Turbo Demodulation with Turbo Decoding: Algorithms, Schedulings, and Complexity Estimation 35
3.1 State of the Art . . . 37
3.2 SISO Demapping Algorithms . . . 38
3.2.1 MAP Demapping Algorithm . . . 39
3.2.2 Max-Log-MAP Demapping Algorithm . . . 39
3.2.3 Parallelism in Turbo Demodulation . . . 40
3.2.3.1 Demapping Metric Level Parallelism . . . 40
3.2.3.2 Demapper Component Level Parallelism . . . 40
3.2.3.3 Turbo Demodulation Level Parallelism . . . 41
3.3 Turbo Demodulation with Turbo Decoding Convergence Speed Analysis . . . 41
3.3.1 TBICM-SSD and TBICM-ID-SSD Error Correction Performance . . . 42
3.3.2 EXIT Chart Block Diagram . . . 42
3.3.3 Effects of Constellation Rotation . . . 44
3.3.4 Effects of Bits-to-Symbol Allocation Scheme . . . 46
3.3.5 Effects of Max-Log-MAP Algorithm . . . 47
3.4 Reducing the Number of Demapping Iterations in TBICM-ID-SSD . . . 48
3.4.1 Proposed TBICM-ID-SSD Scheduling . . . 49
3.4.2 SISO Demapping and SISO Decoding Complexity Evaluation . . . 50
3.4.2.1 SISO Demapping and SISO Decoding Typical Quantization Values 50

3.4.2.2 Complexity Evaluation of the SISO Demapper 51
3.4.2.3 Complexity Evaluation of the SISO Decoder 54
3.4.2.4 Complexity Normalization 55
3.4.3 Discussions and Achieved Improvements 56
3.4.3.1 Complexity Reduction Ratio G_1 57
3.4.3.2 Achieved Improvements 58
3.5 Other Schedulings for TBICM-ID-SSD 58
3.5.1 Scheduling Strategy 59
3.5.2 BER Performance Analysis 59
3.5.3 Complexity Analysis 60
3.6 Complexity Adaptive TBICM-ID-SSD Receiver 62
3.6.1 TBICM-SSD and TBICM-ID-SSD Complexity Expressions 62
3.6.2 Number of Iterations Analysis for Identical Complexity 62
3.6.3 Complexity Analysis for Identical Performance and Achieved Improvements G_3 63
3.6.3.1 Complexity Analysis for a Chosen x 63
3.6.3.2 Complexity Analysis for Different Values of x 66
3.7 Efficient Sizing of Heterogeneous Multiprocessor Receivers for TBICM-ID-SSD . . 68
3.7.1 Generic heterogeneous multiprocessor architecture model 68
3.7.2 Formal representation of the architectural solution space 69
3.7.3 Area optimization 71
3.8 Complexity Reduction of Shuffled Parallel TBICM-ID-SSD 73
3.8.1 Parallel Full Shuffled TBICM-ID-SSD Strategy 73
3.8.2 Simulations and Achieved Improvements 74
3.8.2.1 Performance Simulations for Different Shuffled Turbo Decoding Schemes 75
3.8.2.2 Complexity Analysis and Achieved Improvements G_4 76
3.9 Summary 77

4 Optimized Turbo Equalization with Turbo Decoding: Algorithms, Schedulings, and Complexity Estimation 81
4.1 State of the Art 82
4.2 SISO Equalization Algorithm 83
4.2.1 MMSE Algorithm 84
4.2.2 Demapping 85
4.2.3 Soft Mapping 85
4.2.4 Parallelism in Turbo Equalization 86
4.2.4.1 Symbol Estimation Level Parallelism 86

4.2.4.2 Equalizer Component Level Parallelism 86
4.2.4.3 Turbo Equalization Level Parallelism 87
4.3 Turbo Equalization with Turbo Decoding Convergence Speed Analysis 87
4.3.1 *TEq* and *TEq+TDem* Error Correction Performance 87
4.3.2 EXIT Chart Block Diagram 87
4.3.3 Effects of Constellation Rotation 89
4.3.4 Effects of Feedback to the Equalizer and to the Demapper 90
4.4 Reducing the Number of Equalization Iterations in *TEq* and *TEq+TDem* 91
4.4.1 Proposed *TEq* and *TEq+TDem* Schedulings 91
4.4.2 SISO MMSE Equalizer Complexity Evaluation 94
4.4.2.1 SISO Equalization Typical Quantization Values 94
4.4.2.2 Complexity Evaluation of SISO Equalizer 95
4.4.2.3 Complexity Normalization 97
4.4.3 Discussions and Achieved Improvements 99
4.4.3.1 Complexity Reduction Ratio G_5 99
4.4.3.2 Achieved Improvements 101
4.5 Complexity Adaptive *TEq+TDem* Receiver 101
4.5.1 *TEq+TDem* and *TEq* Performance Simulations 102
4.5.2 Discussions and Achieved Improvements 104
4.6 Summary 106

Conclusions and Perspectives **109**

Résumé en Français **111**

Glossary **121**

Notations **123**

Bibliography **127**

List of publications **135**

List of Figures

1.1 Wireless digital communication system: (a) Transmitter (b) Wireless channel (c) Receiver. . . . 6
1.2 SISO and MIMO transmitter system model for WiMAX standard. . . . 7
1.3 CTC trellis associated with the double-binary CRSC constituent encoder used in WiMAX and DVB-RCS. . . . 8
1.4 CTC encoder used in WiMAX and DVB-RCS, 8-state double binary CRSC code. . . . 9
1.5 Gray mapped QAM16 constellation. . . . 13
1.6 Rotated QAM64 constellation of the DVB-T2 standard with the sub-partitioning technique. . . . 15
1.7 System-level receiver: turbo decoding. . . . 18
1.8 System-level receiver: turbo demodulation with turbo decoding. . . . 19
1.9 System-level receiver: turbo equalization with turbo decoding. . . . 19
1.10 System-level receiver: turbo equalization with turbo demodulation turbo decoding. . . . 19

2.1 System model with TBICM-SSD. . . . 22
2.2 BCJR metrics associated with trellis transitions 26
2.3 Sub-block parallelism with message passing for metric initialization, circular code. . . . 26
2.4 Shuffled turbo decoding. . . . 27
2.5 SISO decoder schemes: (a) Forward-Backward (b) Butterfly (c) Butterfly-Replica. . . . 28
2.6 Classical turbo decoding schedulings: (a) Serial (b) Parallel (c) Shuffled 30
2.7 Shuffled turbo decoding scheduling with overlapping: (a) No Overlapping (b) Overlapping 1 (c) Overlapping 2 31
2.8 BER performance at the output of $\texttt{DEC}_2$ in function of iterations for TBICM-SSD processing in shuffled turbo decoding with butterfly scheme and *Overlapping 1* scheme for different values of δ and scaling factors (2.2.2) for the transmission of 1536 information bits frame over Rayleigh fading channel. QAM16 modulation scheme, R_c=$\frac{1}{2}$ and E_b/N_0=6.25 dB are considered. . . . 32
2.9 BER performance at the output of $\texttt{DEC}_2$ in function of iterations for TBICM-SSD processing in shuffled turbo decoding with butterfly-replica scheme and *Overlapping 1* scheme for different values of δ and scaling factors (2.2.2) for the transmission of 1536 information bits frame over Rayleigh fading channel. QAM16 modulation scheme, R_c=$\frac{1}{2}$ and E_b/N_0=6 dB are considered. . . . 32

2.10 BER performance at the output of $\mathtt{DEC}_2$ in function of iterations for TBICM-SSD processing in shuffled turbo decoding with butterfly scheme and *Overlapping 2* scheme for different values of δ and scaling factors (2.2.2) for the transmission of 1536 information bits frame over Rayleigh fading channel. QAM16 modulation scheme, R_c=$\frac{1}{2}$ and E_b/N_0=6 dB are considered. 33

3.1 Receiver model with TBICM-ID-SSD. 38

3.2 The three typical distinct regions of error correction performance of an iterative process. 42

3.3 BER performance simulations for TBICM-SSD and TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. Different system configurations (QPSK and QAM64 modulation schemes with R_c=$\frac{1}{2}$ and R_c=$\frac{2}{3}$) are considered respectively. 43

3.4 EXIT chart block diagram for turbo demodulation with turbo decoding. 43

3.5 EXIT chart analysis at E_b/N_0= 22 dB of the double-binary turbo decoder for iterations to the QAM64 demapper. $R_c = \frac{4}{5}$ is considered for transmission over Rayleigh fast-fading channel with erasure probability P_ρ=0.15. 45

3.6 EXIT chart analysis at E_b/N_0= 7 dB of the double-binary turbo decoder for iterations to the QPSK demapper. $R_c = \frac{6}{7}$ is considered for transmission over Rayleigh fast-fading channel without erasure. 45

3.7 EXIT chart analysis at E_b/N_0= 14 dB of the double-binary turbo decoder for iterations to the QAM16 demapper. $R_c = \frac{1}{2}$ and Rayleigh fast-fading channel with erasure probability P_ρ=0.15 are considered. 46

3.8 EXIT chart analysis at E_b/N_0= 5.5 dB of the double-binary turbo decoder for iterations to the QAM16 demapper. $R_c = \frac{1}{2}$ and Rayleigh fast-fading channel without erasure are considered. 47

3.9 EXIT chart analysis at E_b/N_0= 5.5 dB of the double-binary turbo decoder for iterations to the QAM16 demapper. $R_c = \frac{1}{2}$ and Rayleigh fast-fading channel without erasure are considered. 47

3.10 BER performance comparison for TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. QAM64 modulation scheme and $R_c = \frac{2}{3}$ are considered. 48

3.11 BER performance comparison for TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel with erasure probability P_ρ=0.15. QAM16 modulation scheme and $R_c = \frac{4}{5}$ are considered. 48

3.12 BER performance comparison for TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. QAM64 modulation scheme and $R_c = \frac{2}{3}$ are considered. 50

3.13 Floating-point vs Fixed-point BER performance comparison for TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel with erasure probability P_ρ=0.15. QPSK and QAM64 modulation schemes are considered respectively for 1, 4, and 8 TBICM-ID-SSD iterations. R_c=$\frac{4}{5}$. 51

3.14 Euclidean distance computation unit 52

3.15 A priori adder computation unit 53

3.16 Minimum Finder for One LLR 54

3.17 Complexity normalization examples: (a) Addition operation (b) Multiplication operation. . . . 56

3.18 BER performance comparison between the original and the new2 TBICM-ID-SSD scheduling as a function of number of iterations for the transmission of 1536 information bits frame over Rayleigh fading channel without erasure. QAM16 modulation scheme, $R_c = \frac{1}{2}$ and E_b/N_0=6 dB are considered. . . . 60

3.19 BER performance comparison between the original and the modified-new2 TBICM-ID-SSD scheduling as a function of number of iterations for the transmission of 1536 information bits frame over Rayleigh fading channel without erasure. QAM16 modulation scheme, R_c=$\frac{1}{2}$ and E_b/N_0=6 dB are considered. . . . 61

3.20 BER performance comparison between TBICM-SSD and TBICM-ID-SSD over iterations for the transmission of 1536 information bits frame over Rayleigh fading channel with and without erasure. Different modulation schemes are code rates are considered . . . 64

3.21 BER performance comparison between TBICM-SSD and TBICM-ID-SSD as a function of number of iterations for the transmission of 1536 information bits frame over Rayleigh fading channel with erasure probability equals to 0.15. QPSK modulation scheme with R_c=$\frac{4}{5}$ and E_b/N_0=9.5 dB are considered. . . . 67

3.22 Complexity reduction over iterations for using TBICM-ID-SSD rather than TBICM-SSD for QPSK with R_c=$\frac{4}{5}$, P_ρ=0.15 and CASE 2. . . . 67

3.23 Generic architecture of the heterogeneous multiprocessor receiver. In this configuration example 2 DemProcs and 4 DecProcs are not used. . . . 69

3.24 Flexible multi-processor hardware area with one extremum for iterative demodulation with turbo decoding. . . . 73

3.25 Parallel full schuffled iterative demapping with turbo decoding. . . . 74

3.26 BER in function of iterations for Butterfly and Butterfly-Replica for different configurations. . . . 75

3.27 Proposal of an adaptive complexity iterative receiver applying turbo demodulation with turbo decoding. . . . 78

4.1 Receiver model with turbo equalization, turbo demodulation, and turbo decoding *TEq+TDem*. . . . 84

4.2 BER performance simulations for *TEq* and *TEq+TDem* for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. Different system configurations (QPSK and QAM64 modulation schemes with 2×2 and 4×4 MIMO SM) are considered. $R_c = \frac{1}{2}$. . . . 88

4.3 EXIT chart block diagram for turbo equalization combined with turbo demodulation and turbo decoding. . . . 88

4.4 *TEq*: EXIT chart analysis at E_b/N_0= 8.25 dB of the double-binary turbo decoder for iterations to the 2×2 SM SISO MMSE equalizer. QAM16 modulation scheme and $R_c = \frac{1}{2}$ are considered for the transmission over Rayleigh fast-fading channel without erasure. . . . 89

4.5 *TEq+TDem*: EXIT chart analysis at E_b/N_0= 8.25 dB of the double-binary turbo decoder for iterations to the 2×2 SM SISO MMSE equalizer and SISO demapper. QAM16 modulation scheme and $R_c = \frac{1}{2}$ are considered for the transmission over Rayleigh fast-fading channel without erasure. 89

4.6 EXIT chart analysis at E_b/N_0=7.25 dB of the double-binary turbo decoder for iterations to the QAM16 demapper and MMSE equalizer. 2×2 MIMO SM and R_c=$\frac{1}{2}$ are considered for the transmission over Rayleigh fast-fading channel without erasure. . 91

4.7 EXIT chart analysis at E_b/N_0=14 dB of the double-binary turbo decoder for iterations to the QAM64 demapper and MMSE equalizer. 4×4 MIMO SM and R_c=$\frac{1}{2}$ are considered for the transmission over Rayleigh fast-fading channel without erasure. . 92

4.8 BER in function of iterations for 2×2 (E_b/N_0=7.25 dB same as for Fig. 4.6) and 4×4 (E_b/N_0=7.75 dB) MIMO SM for QAM16 modulation scheme and R_c=$\frac{1}{2}$. The coded frame size is taken as 768 double-binary symbols. 92

4.9 BER performance comparison for the transmission of 1536 information bits frame over Rayleigh fast-fading channel. 2×2 MIMO SM, QAM16 modulation scheme, and code rate R_c=$\frac{1}{2}$ are considered. 93

4.10 Floating-point vs Fixed-point BER performance comparison for *TEq+TDem* for the transmission of 1536 information bits frame over Rayleigh fast-fading channel with erasure. QPSK with 2×2 MIMO SM and QAM64 with 4×4 MIMO SM are considered respectively for 1, 4, and 8 iterations. R_c=$\frac{1}{2}$. 95

4.11 BER in function of iterations for 2×2 and 4×4 MIMO SM for QPSK and QAM16 modulation schemes. Rayleigh Fast-fading channel, R_c=$\frac{1}{2}$ and N_{CSymb}=768 are considered. 102

4.12 BER in function of iterations for 2×2 and 4×4 MIMO SM for QAM64 and QAM256 modulation schemes. Rayleigh Fast-fading channel, R_c=$\frac{1}{2}$ and N_{CSymb}=768 are considered. 103

4.13 Number of equalization iterations in function of E_b/N_0, to target a specific BER, for QAM16 and QAM64 modulation schemes with different number of antennas. Rayleigh Fast-fading channel, R_c=$\frac{1}{2}$ and N_{CSymb}=768 are considered. 104

4.14 BER performance simulations for *TEq* and *TEq+TDem* for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. QAM64 modulation scheme and 2×2 MIMO SM are considered. $R_c = \frac{1}{2}$. 106

4.15 Proposal of an adaptive complexity MIMO turbo receiver applying turbo demodulation. 106

List of Tables

1.1 Frame sizes (in bytes) specified in WiMAX and DVB-RCS data frame. 8

1.2 Circulation state S_c as function of the length of the data sequence N. 9

1.3 Parity bit Y_1 puncturing patterns for WiMAX and DVB-RCS CTC for different code rates. 11

1.4 Rotation angles in DVB-T2. 14

3.1 Performance loss for different modulation schemes and code rates after 2 omitted demapping iterations over Rayleigh fast-fading channel with and without erasure. . . 49

3.2 SISO demapping and SISO decoding typical quantization values. 51

3.3 SISO demapping and SISO decoding complexity computation summary. 52

3.4 Arithmetic operations normalization in terms of $Add(1,1)$. 55

3.5 SISO demapping and SISO decoding complexity computation summary after normalization. 56

3.6 Reduction in number of operations, read/write access memory comparing "$4IDem\_2EIDec$" to "$6IDem$" for different modulation schemes and code rates. . 58

3.7 Identical complexity (arithmetic operations, or read, or write access memory): Number of required demapping iterations for $x = 6$ and $z = 0$ for different modulation schemes and code rates. 63

3.8 No Erasure channel: Reduction in number of operations, read/write access memory comparing "$3IDem$+$1EIDec$" to "$6IDec$" for different modulation schemes and code rates. 65

3.9 Erasure channel: Reduction in number of operations, read/write access memory comparing "$3IDem$+$0EIDec$" to "$6IDec$" for different modulation schemes and code rates. 65

3.10 TBICM-SSD and TBICM-ID-SSD equivalent number of iterations for QPSK, R_c=$\frac{4}{5}$ and P_ρ=0.15. 66

3.11 Architecture alternatives in function of n. Example for: 200 Mbps, QPSK, R_c=0.5, it_{dem}=it_{dec}=8, $cycles_{dem/symb}$=6, $cycles_{dec/symb}$=1.75 (except for the last iteraton which is equal to 0.75), F_{dec}=F_{dem}=300 MHz 71

3.12 System-level: Reduction in number of arithmetic operations, memory access for using the *B-R* scheme rather than *B* for four different configurations. 77

3.13 System-level: Number of $Add(1,1)$, read and write one-bit memory access operations required in order to process one information symbol for the four configurations using *B* and *B-R* schemes 78

4.1 Performance loss for different modulation schemes, code rates, and number of antennas after one omitted equalization iteration over Rayleigh fast-fading channel without erasure. 94

4.2 MMSE SISO equalization typical quantization values. 95

4.3 MMSE-SISO equalization complexity computation summary with *a priori* information after normalization. 96

4.4 MMSE-SISO equalization complexity computation summary without *a priori* information after normalization. 96

4.5 Reduction in number of operations, read/write access memory comparing $7TEq$+$TDem$ to $6TEq$+$TDem$+$1EIDec$ for 2×2 and 4×4 MIMO SM for different modulation schemes and code rates. 101

4.6 Reduction in number of operations, read/write access memory comparing $6TEq$ to $5TEq$+$1EIDec$ for 2×2 and 4×4 MIMO SM for different modulation schemes and code rates. 101

4.7 Complexity reduction in overall number of arithmetic operations, read, and write memory access for using *TEq*+*TDem* mode rather than *TEq*. $R_c = \frac{1}{2}$. 105

Introduction

RAPIDLY evolving wireless standards use modern techniques such as Turbo codes, Bit Interleaved coded Modulation (BICM), high order Quadrature Amplitude Modulation (QAM) constellation, Signal Space Diversity (SSD), Multi-Input Multi-Output (MIMO) Spatial Multiplexing (SM) and Space Time Codes (STC) with different parameters for reliable high data rate transmissions. Adoption of such techniques in the transmitter can impact the receiver architecture in three ways: (1) the complex processing related to advanced techniques such as turbo codes, encourage to perform iterative processing in the receiver to improve error rate performance (2) to satisfy high throughput requirement for an iterative receiver, parallel processing is mandatory and finally (3) to allow the support of different techniques and parameters imposed, high throughput multi-modes processing elements are required. While translating these requirements on the physical layer of a radio terminal, this can be seen as a flexible, high diversity and high throughput platform which can be configured to the required air interface.

In addition to these technical requirements associated with rapid growth in wireless communication industry, when multiple iterative processes are adopted (e.g. turbo decoding, turbo demodulation, turbo equalization, etc.) real throughput, latency, and power consumption issues appear. In order to handle these issues and to enable the wide adoption of iterative processing, new system-level optimization techniques have to be investigated. One of the main idea in this direction is to analyze the exact convergence impact of each single feedback loop at each functional block and to propose novel schedulings of inner and outer feedbacks which improve the convergence and maximize the overall implementation efficiency by reducing the overall complexity in terms of arithmetic operations and memory accesses with regard to the various communication system parameters.

Objectives

Exploring the ideas presented in the above context constitutes the main objective of this thesis work. The main target is to study the convergence speed and the system-level complexity of advanced wireless communication receivers combining multiple iterative processes. Various communication techniques and system parameters, as specified in emerging wireless communication applications, should be considered. Novel iteration schedulings and iterative receiver configurations should be investigated and proposed to improve the convergence and reduce the overall complexity.

Contributions

1. The first part of this thesis work was focusing on the study of combining turbo demodulation and turbo decoding iterative processes at the wireless receiver. The main contributions accomplished during this part can be summarized as follows:

    - Analyzing the convergence speed of these combined two iterative processes in order to determine the exact required number of iterations at each level. Extrinsic information transfer (EXIT) charts are used for a thorough analysis at different modulation orders and code rates.
    - An original iteration scheduling is proposed reducing two demapping iterations with reasonable performance loss of less than 0.15 dB.
    - Analyzing and normalizing the computational and memory access complexity, which directly impact latency and power consumption, demonstrate the considerable improvements of the proposed scheduling and the promising contributions of the proposed analysis.
    - A complexity and performance study has been done for the two iterative modes TBICM-SSD and TBICM-ID-SSD. It has demonstrated a considerable gain in complexity for using this latter scheduling for low modulation orders (QPSK and QAM16).
    - A throughput study has been done to determine the exact number of decoder and demapper processors required for the two modes TBICM-SSD and TBICM-ID-SSD.
    - A complexity and performance study for the iterative demapping with shuffled turbo decoding receiver applying butterfly and replica-butterfly schemes has been done. It has demonstrated a considerable reduction in complexity for using this latter scheme for all modulations schemes and code rates.

2. The second part of this thesis work has extended the above study to iterative MIMO receivers combining turbo equalization, turbo demodulation, and turbo decoding. The main contributions accomplished in this context can be summarized as follows:

    - Analyzing the convergence speed of these combined three iterative processes in order to determine the exact required number of iterations at each level. EXIT charts are used for a thorough analysis at different number of antennas, modulation orders and code rates.
    - An original iteration scheduling is proposed reducing one equalization iteration for the *TEq* and *TEq+TDem* modes with reasonable performance loss of less than 0.04 dB.
    - Analyzing and normalizing the computational and memory access complexity, which directly impact latency and power consumption, demonstrate the considerable improvements of the proposed scheduling and the promising contributions of the proposed analysis.

- An adaptive complexity MIMO turbo receiver applying turbo demodulation has been proposed. It demonstrated a considerable reduction in complexity for using the adaptive iterative scheduling depending on the system configuration.

Thesis Breakdown

This thesis manuscript is composed of 4 chapters.

Chapter 1 provides the basic requirements of modern wireless digital communication systems in terms of parameters associated with each component of the transmitter. In addition, it presents four iterative processing receivers combining turbo decoding, turbo demodulation, and turbo equalization.

Chapter 2 gives a brief overview of the turbo decoding algorithms, parallelism techniques and different SISO decoding schemes. In addition, it presents three turbo decoding schedulings. Among them, for shuffled turbo decoding, two schemes are proposed by introducing a time delay between the processing of the natural and interleaved constituent component decoders.

Chapter 3 analyzes the convergence speed of the combined two iterative processes namely turbo demodulation and turbo decoding in order to determine the exact required number of iterations at each level. An original iteration scheduling is proposed reducing two demapping iterations with reasonable performance loss of less than 0.15 dB. Furthermore, this chapter illustrates the opposite of what is commonly assumed and proposes a complexity adaptive iterative receiver performing TBICM-ID-SSD depending on the modulation scheme. Moreover, this chapter proposes a flexible multi-processor hardware platform for turbo demodulation with turbo decoding. Platform sizing results analysis demonstrates significant reduction in the area of the iterative receiver. Finally, this chapter demonstrates that the adoption of the butterfly-replica scheme inside the turbo decoder for full shuffled turbo demodulation with turbo decoding can lead to significant complexity reduction.

Chapter 4 analyzes the convergence speed of a complete MIMO receiver applying three iterative processes namely turbo equalization, turbo demodulation and turbo decoding in order to determine the exact required number of iterations at each level to address the ever increasing requirements of transmission quality with lower complexity. An original iteration scheduling is proposed reducing one equalization iteration with maximum performance degradation of 0.04 dB. Normalizing and analyzing the computational and memory access complexity, which directly impact latency and power consumption, demonstrates the considerable gains of the proposed scheduling and the promising contributions of the proposed analysis. In addition, this chapter demonstrates that the adoption of turbo demodulation in the context of turbo equalization combined with turbo decoding can lead to significant complexity reduction for specific system configurations.

CHAPTER

1 Wireless Digital Communication Systems and Iterative Processing

THE objective of this thesis work is to study the convergence speed and the system-level complexity of advanced wireless communication receivers combining multiple iterative processes. In this context, advanced techniques (such as turbo codes, BICM, SSD, MIMO) and various system parameters (such as modulation schemes, code rates, number of antennas) are considered. Each wireless communication standard specifies a sub-set of these techniques with various associated parameters. WiMAX [1], which refers to interoperable implementations of the IEEE 802.16 family of wireless-networks standards approved by the WiMAX Forum, constitutes a representative example. In this standard, four error correction codes (convolutional, Turbo, Low Density Parity Check Codes (LDPC) [2], and block Turbo) are specified and each of them is associated to a large multiplicity of code rates (1/2 to 5/6 for turbo codes) and frame lengths (48 to 4800 bits for turbo codes). BICM technique with three different quadrature amplitude modulation schemes (QPSK, QAM16, and QAM64) is proposed. MIMO Spatial Multiplexing and Space Time Codes with different parameters are specified. Another example is the newly released DVB-T2 standard which couples the BICM with the SSD technique and supports modulation schemes from QPSK to QAM256. Today, there is no one single standard which specifies all these advanced techniques and parameters. However, derived by the trend towards the convergence of radio interfaces, future standards and applications will certainly be more and more complex and rich in terms of specified techniques and parameters.

In this chapter, we give a brief introduction on the techniques and parameters which have been considered in this thesis work. Fundamental concepts of the transmitter components in WiMAX and DVB-RCS standards are presented: channel encoder, BICM [3], constellation mapping, and MIMO transmission. On the receiver side, this chapter will introduce briefly the four iterative receiver configurations which have been studied in this thesis work and which will be developed in subsequent chapters: (1) turbo decoding, (2) turbo demodulation in combination with turbo decoding, (3) turbo equalization in combination with turbo decoding, and (4) turbo equalization in combination with turbo demodulation and turbo decoding.

1.1 Wireless Digital Communication Systems

Starting from Shannon's channel coding theorem [4], the notion of channel capacity was defined as the maximal rate for which information can be transmitted reliably over the channel. Shannon proved that for any channel, there exist families of codes that can achieve arbitrary small probability of error at any communication rate up to the capacity of the channel.

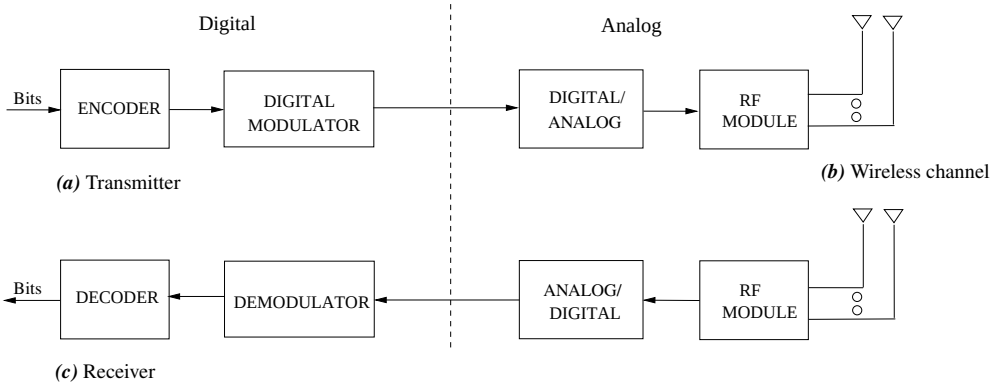


Figure 1.1: Wireless digital communication system: (a) Transmitter (b) Wireless channel (c) Receiver.

All wireless digital communication systems should possess a few key building blocks as shown in Fig. 1.1. Three components namely transmitter, wireless channel and receiver are presented. The construction of the transmitter depends on standards specifications. Depending upon the wireless channel, redundancy and/or diversity is added into the source data to combat against fading and destructive effects of the channel. On the receiver side the received distorted data, composed of source and redundancy, is processed to retrieve the original source.

In last years, different wireless communication standards have been emerged and evolved, such as UMTS [5], 3GPP-LTE [6] for mobile phones, 802.11 (WiFi) and 802.16 (WiMAX) for wireless local and wide area networks, and DVB-RCS, DVB-S2, DVB-T2 for digital video broadcasting.

In this context, WiMAX is one of the emerging wireless networking standards which can be considered rich in various system parameters and application requirements. Additional system parameters such as high modulation orders, high code rates and constellation angles can be found in other standards such as DVB-RCS and DVB-T2 specifications. These standards impose different system parameters for channel coding, BICM interleaving, constellation mapping, and MIMO technology. In the following, we will present some of these specifications. Fig. 1.2 shows the correspondent MIMO transmitter system model.

On the transmitter side, different components are linked together in order to provide immunity against channel effects and to optimally use the available channel bandwidth. Information bits U which are called systematic bits are encoded with a channel encoder. The output codeword C, made up of the source date and parities, is then punctured to reach a desired coding rate R_c.

In order to gain resilience against error bursts, the resulting sequence is interleaved using a BICM interleaver. Punctured and interleaved bits denoted by V are then gray mapped to channel symbols s_q chosen from a 2^M-ary constellation $\mathcal{X}$, M is the number of bits per modulated symbol.

After mapping, single antenna or MIMO transmission is possible. In Single Input Single Output (SISO) transmission, the Signal Space Diversity (SSD) technique [7, 8] can be applied against the fading events. Whereas in MIMO transmission, a Space Time Code (STC) can be used in order to

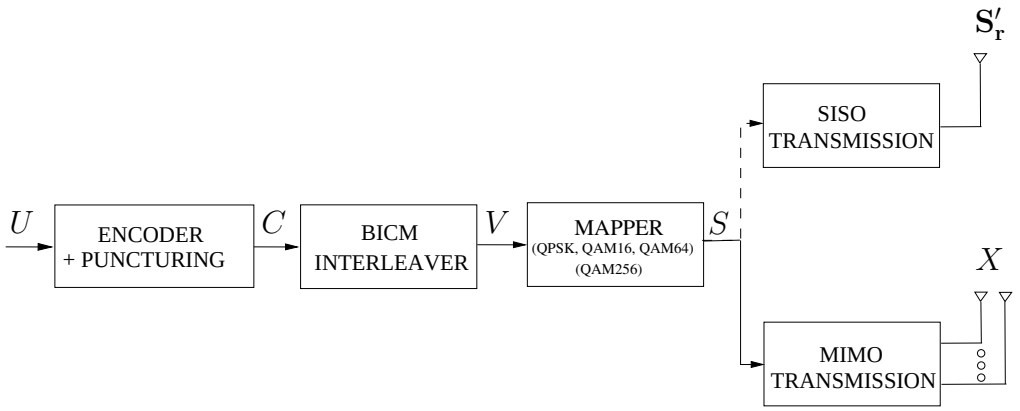


Figure 1.2: SISO and MIMO transmitter system model for WiMAX standard.

provide different features such as time diversity and/or Spatial Multiplexing (SM). Hence, individual symbols of vector X at the output of the MIMO block are transmitted from N_t transmit antennas.

Another technique, called Orthogonal Frequency Division Multiplexing (OFDM), can be used against multi-path fading to counter the Inter Symbol Interference (ISI). This technique is, however, out of the scope of this thesis.

1.1.1 Convolutional Turbo Codes (CTC)

Channel coding is a way of encoding data in a communication channel that adds patterns of redundancy into the transmission in order to enable the receiver to detect and correct transmitted information. Hence, resulted error rate will be lower and information will be transmitted with maximum reliability. Different codes are suggested in wireless communication standards such as Convolutional Turbo Codes (CTC), Low Density Parity Check Codes (LDPC), Convolutional Codes (CC), Reed-Solomon Convolutional Codes (RS-CC), and Block Turbo Codes (BTC).

1.1.1.1 CTC Encoder

In this thesis work, double binary CTC is considered with different coding rates as specified in WiMAX and DVB-RCS. The high performance 8-states double-binary CTC represented by its trellis in Fig. 1.3 and its encoder structure in Fig. 1.4 has been adopted in several studies such as WiMAX and DVB-RCS. The basic concept of turbo codes is that the information sequence is encoded twice, with an interleaver between the two encoders serving to make the two encoded data sequences approximately statistically independent from each other. The output of the double-binary CTC encoder consists of the two systematic bits c_p and c_{p+1} and the four parity bits c_{p+2}, c_{p+3}, c_{p+4} and c_{p+5}.

The encoder is fed with data blocks of N bytes which are grouped into $4N$ bit-couples (or double-binary symbols). The number of bytes per block N for the WiMAX standard [1] and for DVB-RCS [9] are shown in Table 1.1. For WiMAX, seventeen frame sizes are specified raging from 6 bytes to 600 bytes. For DVB-RCS, twelve frame sizes are specified raging from 12 bytes to 216 bytes, including a 53 byte frame compatible with Asynchronous Transfer Mode (ATM) and a 188 byte frame compatible with MPEG-2.

Due to complexity/performance issues for turbo decoding, small component codes have been chosen. The chosen constraint length for the constituent encoder is equal to 3 as shown in Fig. 1.4,

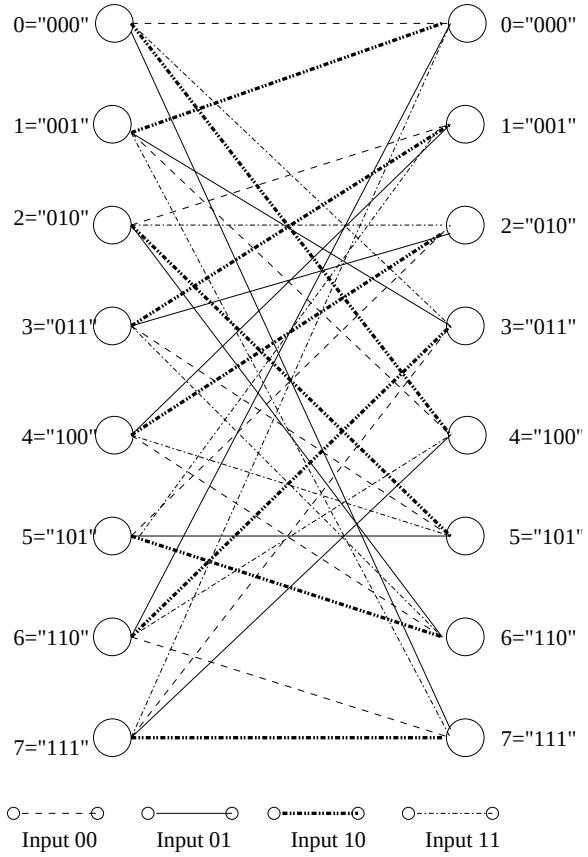


Figure 1.3: CTC trellis associated with the double-binary CRSC constituent encoder used in WiMAX and DVB-RCS.

WiMAX	6	9	12	18	24	27	30	36	45	48	54	60	120	240	360	480	600
DVB-RCS	12	16	53	55	57	106	108	110	188	212	214	216					

Table 1.1: Frame sizes (in bytes) specified in WiMAX and DVB-RCS data frame.

hence 2^3=8 states exist. In fact, the coding gain grows almost linearly with the code memory while the complexity of the decoding grows exponentially. Furthermore, the CTC encoder of Fig. 1.4 calls for two techniques in turbo coding:

1. parallel concatenation of two identical Circular Recursive Convolutional component Codes (CRSC) with generators (in octal notation) 15 (recursion), 13 (redundancy Y_1), 11 (redundancy Y_2): In fact, the adoption of circular coding avoids the degradation of the spectral efficiency of the transmission when forcing the value of the encoder state at the end of each encoding frame by the addition of tail bits [10]. CRSC is an adaptation of the so called *tail-biting* technique to Recursive Convolutional Codes (RSC). CRSC ensures that at the end of the encoding operation, the encoder retrieves the initial state, so that data encoding may be represented by a circular trellis. Furthermore, CRSC are less susceptible to puncturing and sub-optimal decoding algorithm [11].

   The value of the circulation state S_c (state of the encoder) depends on the contents of the sequence to encode, $0 \leq S_c \leq 7$. Determining S_c requires a pre-encoding operation. First, the encoder is initialized in the *all zero* state, then the data sequence is encoded once leading to a final state S_N. S_c value can be computed from the expression $S_c=(1+G)^{-1}S_N$ where G is the generator matrix of the considered code. After that, data is encoded for the second time starting from state S_c.

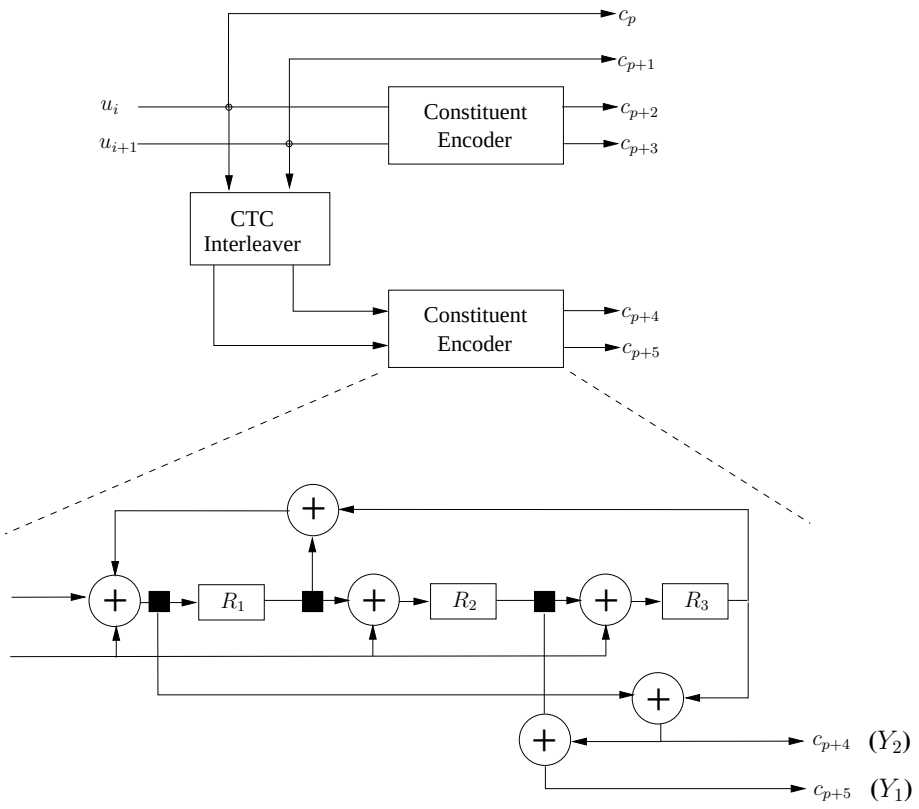


Figure 1.4: CTC encoder used in WiMAX and DVB-RCS, 8-state double binary CRSC code.

Generally, a Look Up Table (LUT) is used to find the circulation state of the encoder Hence, according to Table 1.2 and to the length N of the data sequence, S_c is found.

N modulo 7	S_N=0	S_N=1	S_N=2	S_N=3	S_N=4	S_N=5	S_N=6	S_N=7
1	0	6	4	2	7	1	3	5
2	0	3	7	4	5	6	2	1
3	0	5	3	6	2	7	1	4
4	0	4	1	5	6	2	7	3
5	0	2	5	7	1	3	4	6
6	0	7	6	1	3	4	5	2

Table 1.2: Circulation state S_c as function of the length of the data sequence N.

2. double-binary convolutional codes: Information bits U are regrouped into symbols u_i consisting of ∇=2 bits. Compared to single-binary turbo codes, double-binary turbo codes have several advantages.

   - Reducing the sensitivity to puncturing [12]: since the rate $\frac{1}{3}$ of double-binary RSC encoder produces two parity streams, most of the code rates can be obtained by simply ignoring one of these parity streams and puncturing the other (if necessary). Ignoring one of the two parities results in a new RSC encoder with a single parity stream. This single parity stream is less punctured compared to similar single-binary convolutional RCS encoders, which results in less sensitivity to puncturing.

- Reducing the correlation effects between component decoders: this feature leads to improved convergence [13], which leads to less significant degradation in performance for the simplified versions of the Maximum A Posteriori (MAP) algorithms (0.15 db for double-binary instead of 0.3-0.4 db for binary turbo codes).

The CTC trellis associated with the double-binary CRSC constituent encoder used by WiMAX and DVB-RCS is shown in Fig. 1.3.

1.1.1.2 CTC Interleaver

From implementation perspective, the simplest way to achieve interleaving in a block is to adopt uniform or regular interleaving: data is written in line wise and read in column wise in a rectangular matrix. This kind of permutation behaves very well towards error patterns with short weights, but is very sensitive to square or rectangular error patterns, as explained in [14] [15]. Generally, to make bigger the distances given by rectangular error patterns, non-uniformity is introduced in the permutation relations. The disorder introduced with non-uniformity affect the diffusing properties for short error patterns weights.

However, the CTC encoder adopted in the WiMAX standard uses a different type of CTC interleaver, called Almost regular permutation (ARP). It can be described through the following two steps:

Let the sequence A_0=[$(u_0, u_1)_0$, $(u_0, u_1)_1$, . . ., $(u_0, u_1)_j$, . . ., $(u_0, u_1)_{L-1}$] be a frame of size $2L$ bits (or L couples of bits, or L double-binary symbols).

Step 1: Switch alternate couples
for j = 0, . . ., $L-1$
if (j mod 2 = 1) switch the couple $(u_0, u_1)_j \Rightarrow (u_1, u_0)_j$
This step results in the sequence A_1 = [$(u_0, u_1)_0$, $(u_1, u_0)_1$, . . ., $(u_1, u_0)_j$, . . .] = $[A_1(0), A_1(1), \ldots, A_1(j), \ldots]$.

Step 2: Switch between couples
for j = 0, . . ., $L-1$
use the function $\prod(j)$ which provides the interleaved address of each couple of index j from the sequence A_1

$$\prod(j) = (P_0 \times j + P + 1)\ mod\ L$$

with

$$\begin{aligned} P &= 0 && \text{if } j\ mod\ 4 = 0 \\ P &= \frac{L}{2} + P_1 && \text{if } j\ mod\ 4 = 1 \\ P &= P_2 && \text{if } j\ mod\ 4 = 2 \\ P &= \frac{L}{2} + P_3 && \text{if } j\ mod\ 4 = 3 \end{aligned}$$

where the parameters P_0, P_1, P_2 and P_3 depend on the frame size and are specified in the corresponding standard [1, 9].

It is worth to note that this ARP interleaver is well suited for hardware implementation and presents a collision-free property for certain level of parallelism.

1.1.1.3 CTC Puncturing

Each one of the two component codes inside the convolutional turbo encoder is producing a systematic output which is equivalent to the original information sequence, as well as two streams of parity

information. The two parity sequences can then be punctured before being transmitted along with the original information sequence to the decoder. Note that systematic bits are not punctured, since this degrades the performance of the code more dramatically than puncturing parity bits. This puncturing of the parity information allows a wide range of coding rates to be realized. The code rates that could be achieved for WiMAX are 1/2, 2/3, 3/4 and 5/6. Regarding DVB-RCS, code rates are 1/3, 2/5, 1/2, 2/3, 3/4, 4/5, 5/6, and 6/7. The puncturing patterns are identical for both component codes. Note that the communication standards specify for each frame length a set of supported code rates R_c. For R_c=1/3 no puncturing is required, 1 source pair of 2 bits is generating 6 coded bits. For R_c=2/5, both encoders maintain all the Y_1 (Fig. 1.4) but delete odd-indexed Y_2. For the other considered code rates, Table 1.3 shows the puncturing patterns for parity bit Y_1. Value "1" means keeping the corresponding parity bit, and "0" means deleting the corresponding parity bit. Parity bit Y_2 is always punctured (not considered). Taking the example of R_c=4/5, only every fourth Y_1 is maintained.

Code rate R_c	Index of the double-binary symbol											
	0	1	2	3	4	5	6	7	8	9	10	11
1/2	1	1										
2/3	1	0	1	0								
3/4	1	0	0	1	0	0						
4/5	1	0	0	0	1	0	0	0				
5/6	1	0	0	0	0	1	0	0	0	0		
6/7	1	0	0	0	0	0	1	0	0	0	0	0

Table 1.3: Parity bit Y_1 puncturing patterns for WiMAX and DVB-RCS CTC for different code rates.

1.1.2 Bit-Interleaved Coded Modulation (BICM)

The BICM principle currently represents the state-of-the-art in coded modulations over fading channels. It was first introduced by Zehavi in [16] and later on formalized by Caire *et al.* in [3]. It is a flexible modulation/coding scheme which allows the designer to choose a modulation constellation independently of the coding rate. This is due of the output of the channel encoder and the input to the modulator which are separated by a bit-level interleaver. Coded binary bits are dispersed on different modulated symbols after modulation. By doing this, bits from different coded symbols will be affected by different fading effects. Hence, the error correction capability of the decoder at the receiver side will increase.

In order to increase spectral efficiency, BICM can be combined with high-order modulation schemes such as quadrature amplitude modulation (QAM) or phase shift keying. BICM is particularly well suited for fading channels, and it only introduces a small penalty in terms of channel capacity when compared to the coded modulation capacity for both additive white Gaussian noise (AWGN) and fading channels. Additionally, applying iterative demodulation in this context (BICM-ID) improves the system performance. Similarly, BICM coupled with turbo equalization can be applied to provide excellent error rate performance results.

Regarding the WiMAX standard, encoded data bits are interleaved by a block interleaver with a block size corresponding to the number of coded bits. The interleaver is made of two steps [1]:

- The first permutation ensures that adjacent coded bits are mapped onto non-adjacent modulated symbols.

- The second permutation insures that adjacent coded bits are mapped alternately onto less or more significant bits of the constellation, thus avoiding long runs of bits of low reliability.

Let M be the number of coded bits per modulated symbol, i.e., 2, 4, or 6 for QPSK, QAM16, or QAM64, respectively. Let s=$M/2$. Within a block of M bits at transmission, let k be the index of the coded bit before the first permutation, m_k be the index of that coded bit after the first and before the second permutation and let j_k be the index after the second permutation, just prior to modulation mapping, and d be the modulo used for the permutation.
The first permutation is defined by equation 1.1.

$$m_k = (M/d) \cdot k_{mod(d)} + floor(k/d) \tag{1.1}$$

The second permutation is defined by equation 1.2.

$$j_k = s \cdot floor(m_k/s) + (m_k + M - floor(d.m_k/M))_{mod(s)} \tag{1.2}$$

where $k = 0, 1, \ldots, M$ and $d = 16$.

The de-interleaver function, which performs the inverse operation, is also defined by two permutations [1]. Within a received block of M bits, let j be the index of a received bit before the first permutation, m_j be the index of that bit after the first and before the second permutation, and let k_j be the index of that bit after the second permutation, just prior to delivering the block to the decoder. The first permutation is defined by equation 1.3.

$$m_j = s \cdot floor(j/s) + (j + floor(d \cdot j/M))_{mod(s)} \tag{1.3}$$

The second permutation is defined by equation 1.4.

$$k_j = d \cdot m_j - (M - 1)floor(d \cdot m_j/M) \tag{1.4}$$

where $k = 0, 1, \ldots, M - 1$ and $d = 16$.

The first permutation in the de-interleaver is the inverse of the second permutation in the interleaver, and conversely.

1.1.3 Mapping

After the BICM interleaving block, the data bits are entered serially to the constellation mapper. The modulation process, in a digital communication system, maps a sequence of binary data onto a set of corresponding signal waveforms. These waveforms may differ in either amplitude or phase or in frequency, or some combination of two or more signal parameters.

1.1.3.1 Quadrature Amplitude Modulation (QAM)

For WiMAX and DVB-RCS, only the quadrature amplitude modulation is used. WiMAX supports three different modulation schemes: QPSK, QAM16, and QAM64. Meanwhile, DVB-RCS supports four modulation schemes including the ones for WiMAX and the QAM256 scheme. Each modulation constellation is scaled by a number c, such that the average transmitted power is unity, assuming that all symbols are equally likely. The value of c is $\sqrt{2}$, $\sqrt{10}$, $\sqrt{42}$, and $\sqrt{42}$ for QPSK, QAM16, QAM64, and QAM256 respectively.

1.1.3.2 Gray Mapping

In [17], the authors have investigated different mapping techniques suited for BICM-ID and QAM constellations. They proposed several mapping schemes providing significant coding gains. The Major Set Partitioning (MSP) labeling has shown the best performances with convolutionally coded BICM.

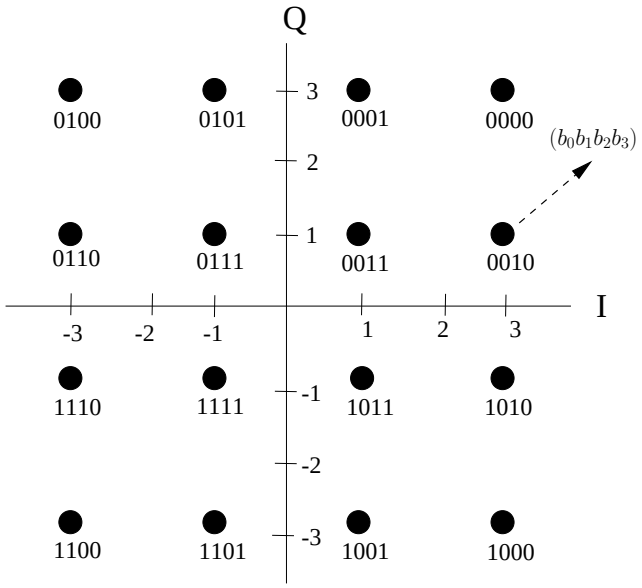


Figure 1.5: Gray mapped QAM16 constellation.

In this work, only the Gray mapping will be used. Adjacent constellation symbols differ by only one bit. In fact, the Gray mapping scheme provides the best performances for BICM with and without iterative demapping when turbo code is used [18]. For this mapping, a QAM scheme is reduced to two independent Pulse Amplitude Modulations (PAM) signals, each carrying $M/2$ bits. Fig. 1.5 shows the QAM16 constellation with Gray coded mapping.

1.1.3.3 Signal Space Diversity (SSD)

BICM coupled with SSD has been extensively studied for single carrier systems, e.g. in [7, 8] and the references therein. In addition, authors in [19] have presented the SSD technique intended to improve the performance of Turbo BICM and BICM-ID (TBICM-ID) over non Gaussian channels. The SSD technique consists of a rotation of the constellation followed by a signal space component interleaving. It has shown additional error correction at the receiver side in an iterative processing scenario.

In fact, constellation rotation enables to exploit higher code rates and to solve potential problems in selective channels while keeping good performances. It has been proposed for all constellation orders of QAM. The performance gain obtained when using a rotated constellation $\mathcal{X}_r$ depends on the choice of the rotation angle. In this regard, a thorough analysis has been done for the 2nd-generation terrestrial transmission system developed by the DVB Project (DVB-T2) which adopted the rotated constellation technique. A single rotation angle [20] has been chosen for each constellation size independently of the channel type. Using these angles, and with LDPC code, gains of 0.5 dB and 6 dB were shown for Rayleigh fading channel without and with erasure respectively for high code

rate [20]. These angles are presented in Table 1.4 and are adopted in this work. This rotation does not change neither the distances between the constellation points nor the distance to the origin. Hence, no modification in transmission power or bandwidth is required.

Modulation	Rotation Angle(Φ)[degrees]
QPSK	29
16-QAM	16.8
64-QAM	8.6
256-QAM	3.6

Table 1.4: Rotation angles in DVB-T2.

Combining constellation rotation with signal space component interleaving leads to significant improvement in performance over fading and erasure channels. It increases the diversity order of a communication system without using extra bandwidth. When a constellation signal is submitted to a fading event, its in-phase component I and quadrature component Q fade identically and suffers from an irreversible loss. A means of avoiding this loss involves making I and Q fade independently while each carrying all the information regarding the transmitted symbol. By inserting an interleaver (a simple delay for uncorrelated fast-fading channel) between the I and Q channels, the diversity order is doubled.

1.1.3.4 Constellation Sub-Partitioning Technique

At the receiver side, the demapping complexity increases significantly with high modulation orders. Complexity reductions in this level can be achieved for medium and high constellation sizes (as for QAM64, QAM256) by applying the sub-partitioning technique of the constellation as presented in [21]. This technique reduces the search of closest constellation point complexity order for QAM from 2^M to $(2^{\frac{M-2}{2}}+1)^2$.

In fact for the QAM256 case, 256 euclidean distances should be computed for the classical demapping. Hence, 512 arithmetic multiplications (256 for the in-phase component I and 256 for the quadrature component Q) are required. In order to reduce this number of euclidean distance computations, the sub-partitioning technique is proposed [21] based on the constellation division into four sub-regions. The choice and the sizing of these sub-regions follow two rules:

1. For a given received signal, the partitioning of the gray mapped constellation into sub-regions can be accorded to the sign of the received channel observation.

2. The corresponding sub-region is dimensioned such as, for any point of the selected region, it will contain all the points that differ only by one bit of the considered point. Consequently, any sub-region should include the closest two points with values 0 and 1 for every estimated output computation.

Fig. 1.6 shows the rotated QAM64 constellation adopted in DVB-T2. Each point of this constellation carries six bits. When the I and Q components of the received signal are positive, the selected sub-region is the red one. The other three sub-regions correspond to the other three possible I and Q sign combinations. In this case, the number of Euclidean distance computations has been reduced from 64 to 25. Moreover, for the QAM256 case, it can be reduced from 256 to 81.

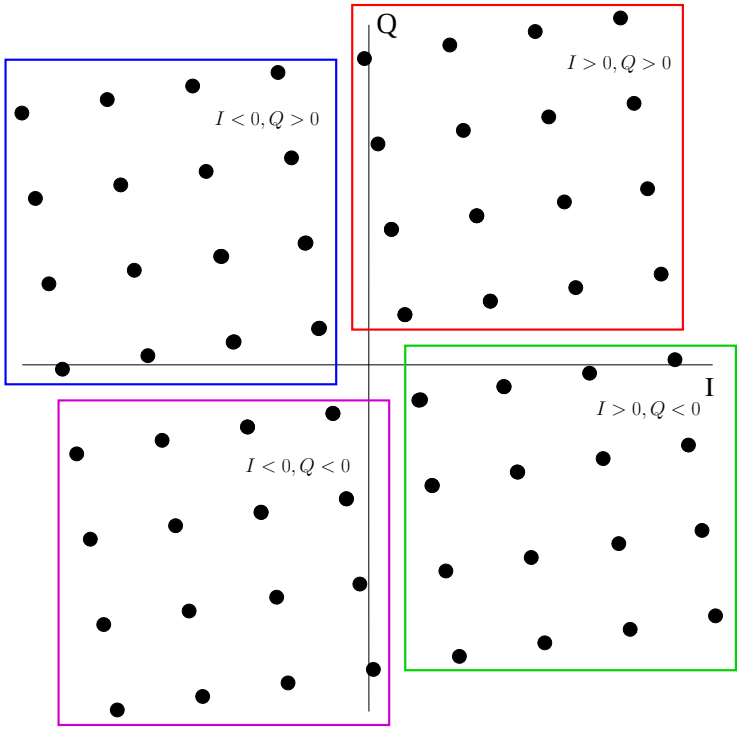


Figure 1.6: Rotated QAM64 constellation of the DVB-T2 standard with the sub-partitioning technique.

1.1.3.5 Bits-to-Symbol Allocation Technique

Restricted to Gray mapping (mandatory due to the use of a turbo code), QAM schemes offer different types of bit protection in each component axis depending on the position of the allocated bit within the transmitted symbol and the order of the modulation. For example, the Gray mapped QAM16 of Fig. 1.5 offers two levels of bit protection. b_0 and b_1 can provide more protection than b_2 and b_3 due to the position of their respective decision regions. Bit error rates at those latter are approximately the double than the others. In [22], the most protected bit positions were allocated to parity bits. In [23], these positions were allocated to systematic bits. The latter allocation method associated with a turbo code outperforms the former in the waterfall region due to the fact that systematic bits are used in both component decoders compared to parity bits that are used only in one. Nevertheless, lower error floors are achieved when parity bits are better protected.

1.1.4 MIMO Techniques

MIMO techniques are being widely adopted in emerging wireless communication systems. The use of the MIMO technology increases the diversity, improves the reception and allows for a better rate of transmission. MIMO techniques can be divided into two main categories: Spatial Multiplexing (SM) and Space Time coding (STC).

1.1.4.1 Space-Time Coding (STC)

The basic idea of Space-Time Coding (STC) is to create redundancy or correlation between symbols transmitted on the spatial and temporal dimensions. A space-time code is characterized by its rate,

the order of diversity and its coding gain. The rate of space-time code is equal to the ratio between the number of symbols transmitted and their corresponding number of transmission periods. The diversity order corresponds to the number of independent channels at the receiver side. Finally, the coding gain is the gain made by the coded system, in terms of performance, compared to non-coded system. A space-time code is said to be full rate when the rate is equal to the number of antennas at the transmitter. A space-time code is said to have maximum diversity when it is able to exploit a diversity equal to $N_t \times N_r$. N_t and N_r represent respectively the number of transmitted and received antennas.

1.1.4.2 Spatial Multiplexing (SM)

The 802.16 WiMAX specification supports the MIMO technique of spatial multiplexing, also known as Transmit Diversity rate = 2 (aka Matrix B in the 802.16 standard). Instead of transmitting the same bit over two antennas, this method transmits one data bit from the first antenna, and another bit from the second antenna simultaneously, per symbol. As long as the receiver has more than one antenna and the signal is of sufficient quality, the receiver can separate the signals. However, with two transmit antennas and two receive antennas, data can be transmitted twice as fast as compared systems using Space Time Codes with only one receive antenna. For the rest of this thesis, only the SM technique will be considered.

1.1.5 Channel Models

The characteristics of wireless signal changes as it travels from the transmitter antenna to the receiver antenna. These characteristics depend upon the distance between the two antennas, the paths taken by the signal, and the environment (buildings and other objects) around the path. The profile of received signal can be obtained from that of the transmitted signal if we have a model of the medium between the two. This model of the medium is called channel model.

A wireless channel is typically modeled with additive noise and multiplicative fading. The noise is added to the received signal at the input of the receiver whereas the fading influences the transmitted signal while passing through the channel. In this thesis, the Additive White Gaussian Noise (AWGN) is considered while a fading coefficient has a Rayleigh distribution. In addition to these two main factors, there are other parameters which are used to model the channel. One of the them is the presence of multipath delays comparable to the time delay between two transmitted symbols. This situation gives rise to ISI and the channel is called frequency selective. A second parameter used in characterizing the channel is the variation of the channel in time, also referred as selectivity in time.

1.1.5.1 Frequency Selective Channel

Frequency selectivity fading multipath channels are often encountered in wireless communication systems. To combat ISI on such channels, receivers use various equalization techniques. A channel is called frequency selective when its frequency response is not perfectly flat because of echoes and reflections generated during the transmission. In a multipath situation, this signals arriving along different paths will have different attenuations and delays and they might add at the receiving antenna either constructively or destructively. For a single antenna transmission system, this channel can be described by the equation:

$$y(t) = \sum_{i=0}^{L-1} h_i(t)x(t-i) + w(t) \tag{1.5}$$

where y and x represent respectively the received and transmitted signals. w is AWGN. L is the number of paths taken by the transmitted signal, reflecting the temporal dispersion of the channel during symbol transmission period. h_i represents the fading of the path i applied to a signal transmitted at time t-i. For the rest of this thesis, we will consider the channel as frequency non-selective for both single antenna and MIMO systems. This is due to the reason that emerging wireless standards use OFDM technique to avoid ISI caused by frequency selectivity of the channel. Hence for single antenna case, equation (1.5) becomes:

$$y(t) = h(t)x(t) + w(t) \tag{1.6}$$

Similarly for MIMO systems with N_t transmit antennas and N_r receive antennas, the relation between channel, transmitted symbols and received symbols is given by the expression below:

$$Y = HX + W \tag{1.7}$$

where

$$\begin{aligned} Y &= [y_1, \ldots, y_{N_r}]^T \in \mathbb{C}^{N_r \times 1} \\ X &= [x_1, \ldots, x_{N_t}]^T \in \mathbb{C}^{N_t \times 1} \\ W &= [\omega_1, \ldots, \omega_{n_R}]^T \in \mathbb{C}^{n_R \times 1} \\ H &= \begin{bmatrix} h_{11} & \cdots & h_{1N_t} \\ \vdots & \ddots & \vdots \\ h_{N_r1} & \cdots & h_{N_rN_t} \end{bmatrix} \end{aligned}$$

where Y and X represent respectively the received and transmitted symbol vectors. W represents the AWGN vector. H is the channel matrix whose element h_{ij} represents the fading coefficient that characterizes the relation between the i^{th} receive antenna and j^{th} transmit antenna.

1.1.5.2 Time Selective Channel

High relative mobility between the transmitter and the receiver causes the transmission channel to change rapidly in time, which is referred to as the time selectivity of the channel. This selectivity characterizes 3 types of channels:

- The *fast-fading channel*: varies at each symbol period.
- The *quasi-static channel*: remains constant during the transmission of a frame.
- The *block fading channel*: remains constant during the transmission of a given number of sub-blocks of the frame.

1.1.5.3 Double Selective Channel

The channel with both the frequency selectivity and the time selectivity is called doubly selective channel in wireless communications.

1.2 Iterative (Turbo) Processing

Iterative processing is widely adopted nowadays in modern wireless receivers. In fact, it is fair to say that turbo codes have caused a paradigm shift in communication theory. The idea of passing information back and forth between different components in a receiver (so-called iterative processing or turbo processing) has become prevalent in state-of-the-art receiver design. Extension of this principle with an additional iterative feedback loop to the demapping and equalization functions has proven to provide substantial error performance gain in many works in the state of the art.

1.2.1 Turbo Decoding

Fig. 1.7 shows the classical turbo decoding receiver. Exploiting the redundancy and diversity added to source data in the transmitter, the receiver tries to remove the channel effects in order to retrieve the original source data. Each constituent decoder unit processes the data once and then passes the information to the next unit. This constitutes a turbo decoding iteration.

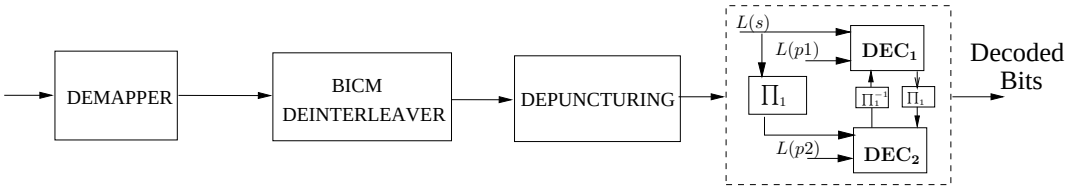


Figure 1.7: System-level receiver: turbo decoding.

1.2.2 Turbo Demodulation with Turbo Decoding

Extension of the principle above with an additional iterative feedback loop to the demapper can provide significant error performance gains at the cost of increased complexity per iteration. Feedback path exist in addition to forward path, through which, the turbo decoder can send soft output information to the demapper unit iteratively. Fig. 1.8 shows the turbo demodulation receiver applying turbo decoding. Many iteration schedulings can be found in the state of the art for this receiver. One of them [19] is to execute only one turbo decoding iteration for each demapping iteration.

1.2.3 Turbo Equalization with Turbo Decoding

The traditional methods of data protection used in error correction code do not work well when the channel over which the data is sent introduces additional distortions in the form of ISI. When the channel is frequency selective or for other reasons is dispersive in nature, the receiver will need to compensate the channel effects. Such channel compensation is typically referred to as channel equalization. An iterative decoder structure combining channel equalization and turbo decoding is presented in Fig. 1.9. At each iteration extrinsic information from the equalizer is fed into the turbo

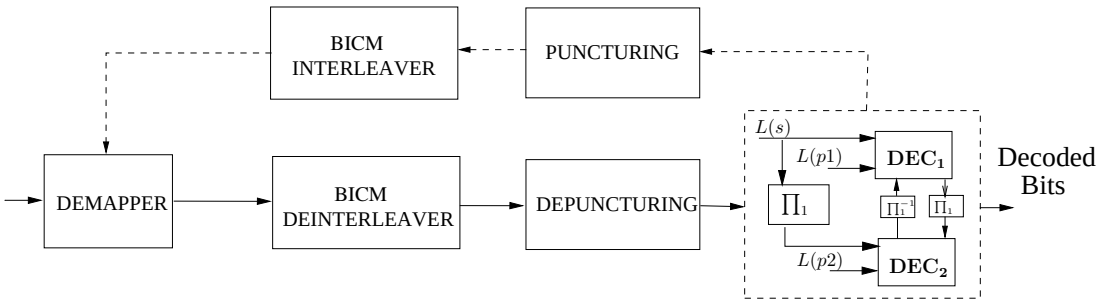


Figure 1.8: System-level receiver: turbo demodulation with turbo decoding.

decoder. The resulted turbo decoder extrinsic information is fed back to the channel equalizer. In fact, the concept of turbo equalization was first introduced in [24] to combat the detrimental effects of ISI for digital transmission protected by convolutional code. Many iteration schedulings for iterative equalization and turbo decoding can be found in the state of the art. One of them [25] is to execute only one turbo decoding iteration for each equalization iteration.

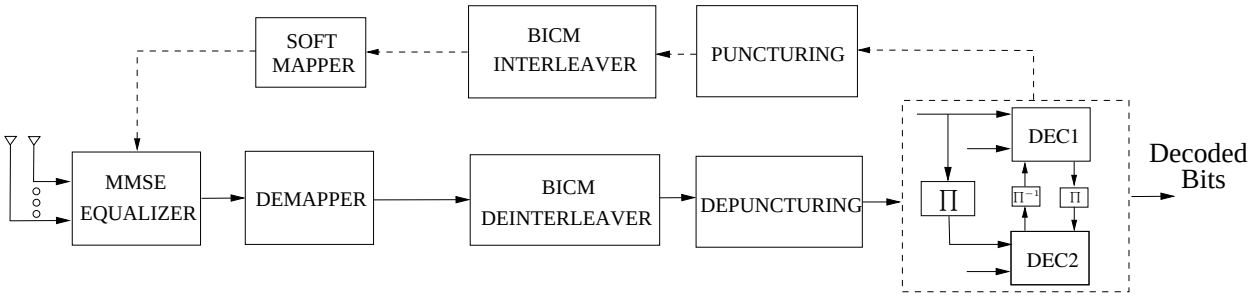


Figure 1.9: System-level receiver: turbo equalization with turbo decoding.

1.2.4 Turbo Equalization with Turbo Demodulation and Turbo Decoding

In the same context of the subsection above, additional feedback from the turbo decoder to the demapper can be applied. The considered system receiver will combine turbo equalization, turbo demodulation, and turbo decoding. To the best of our knowledge, the convergence speed analysis of this turbo receiver applying turbo decoding (Fig. 1.10) has never been done before. A preliminary work can be found in [26] where the authors have presented a similar receiver using a convolutional code.

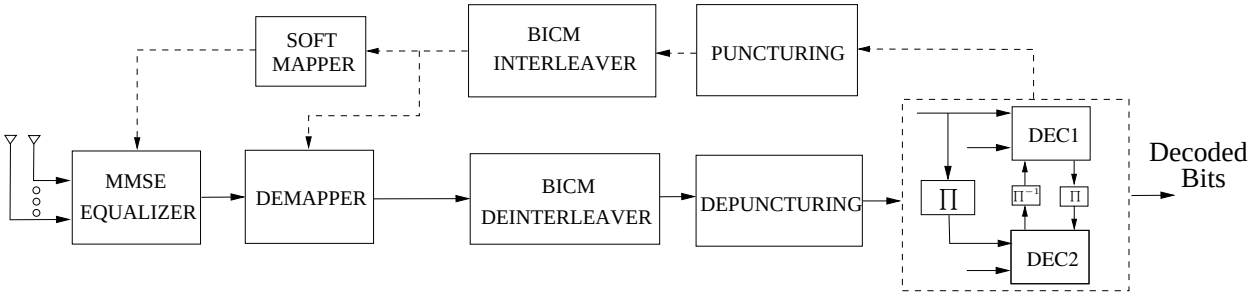


Figure 1.10: System-level receiver: turbo equalization with turbo demodulation turbo decoding.

1.3 Summary

In this first chapter, a brief introduction on advanced techniques and various system parameters used in emerging wireless communication standards and considered in this thesis work have been presented. This includes: (a) turbo codes as specified in the WiMAX standard with a large multiplicity of code rates (1/2 to 5/6), (b) BICM technique with different quadrature amplitude modulation schemes (QPSK, QAM16, QAM64, and QAM256), (c) SSD technique with different rotation angles as specified in the DVB-T2 standard, and (d) MIMO transmission with different number of antennas.

Furthermore, an effort has been made to describe briefly the principles of four turbo receivers combining turbo decoding, turbo demodulation and turbo equalization. Analyzing these iterative receivers in order to propose novel iteration schedulings which improve the convergence and reduce the overall complexity is the object of the subsequent chapters.

CHAPTER

2 Turbo Decoding: Algorithms and Schedulings

IN the previous chapter, the basic requirements of advanced wireless digital communication systems have been summarized. Various parameters associated with the transmitter, the channel, and the receiver were discussed.

For the turbo decoding receiver, this chapter gives a brief overview of the convolutional decoding algorithms. The exact MAP and approximated Max-Log-MAP decoding algorithms are presented. Turbo decoding parallelism techniques are then cited and classified in three different levels.

Moreover, this chapter gives a complete description of the different SISO turbo decoding schemes. Furthermore, it presents the serial, parallel, and shuffled turbo decoding schedulings. Among them, for the shuffled turbo decoding scheduling, two schemes are proposed by introducing a time delay between the processing of the natural and interleaved constituent decoder components.

2.1 State of the Art

Advanced wireless communication standards impose the use of modern techniques to improve spectral efficiency and reliability. Among these techniques Turbo Codes with various code rates are frequently adopted. Moreover, it was shown that BICM [3] offers a significant improvement in error correcting performance for coded modulation over Rayleigh fading channels compared to the previously existing techniques like Trellis Coded Modulation (TCM) [27].

In [28], the convolutional code was replaced by a turbo code (TBICM). This latter has the advantage of separating the code from the modulation without significant loss over Gaussian channels compared to the so-called "turbo trellis-coded modulation" (TTCM) [29]. The TBICM scheme features high coding diversity (well suited for fading channels), high flexibility as well as design and implementation simplicity, while maintaining good power efficiency.

In [19], authors have presented the SSD technique intended to improve the performance of TBICM over non Gaussian channels. The proposed technique consists of a rotation of the constellation followed by a signal space component interleaving. It has shown additional error correction at the receiver side.

2.2 SISO Decoding Algorithms

Following the demapping function at the receiver side, the turbo decoding algorithm is applied. If we look at the history of decoding algorithms, several ones have been proposed to decode a convolutional code. The initial algorithms are presented by Fano [30] and Viterbi [31] which have binary inputs and outputs. The Viterbi algorithm which is better than the other was later modified to accept the soft inputs to improve the decoding [32]. The Soft Output Viterbi Algorithm (SOVA) [33] takes the soft input and provides the soft output as well. Among the SISO algorithms, the Cock-Bahl-Jelinek-Raviv (BCJR) [34] also called MAP (Maximum A Posteriori) or forward backward algorithm, is the optimal decoding algorithm which calculates the probability of each symbol from the probability of all possible paths in the trellis between initial and final states. The MAP algorithm is more complex compared to SOVA. At high Signal-to-Nose Ratio (SNR), the performance of SOVA and MAP are almost the same. However, at low SNR, the MAP algorithm is superior to SOVA by 0.5 dB or more [35].

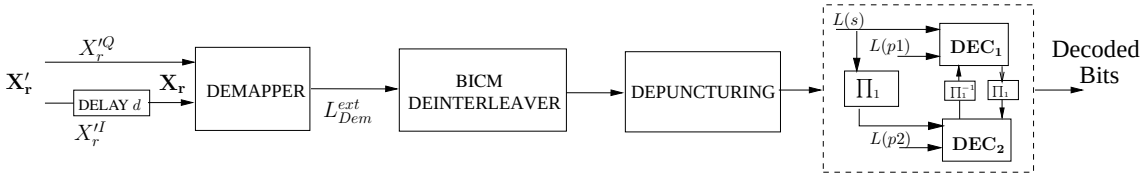


Figure 2.1: System model with TBICM-SSD.

The considered receiver model applying turbo decoding is shown in Fig. 2.1. The channel model considered is a frequency non-selective memoryless channel. The received discrete time baseband complex signal can be written as:

$$x'_{r,q} = h_q.s'_{r,q} + n_q \tag{2.1}$$

where h_q is the Rayleigh fast-fading coefficient. n_q is a complex white Gaussian noise with spectral density $N_0/2$ in each component axes.

At this side, Complex received symbol $\mathbf{X'_r}$ has its Q-components re-shifted resulting in $\mathbf{X_r}$. Decoder extrinsic log-likelihood ratio L_{Dec}^{ext} is calculated for each coded bit per decoding iteration. Finally, the hard decisions are generated at the last iteration.

2.2.1 MAP Decoding Algorithm

The MAP algorithm is optimal but computationally complex SISO algorithm. For each source symbol d_k=u_iu_{i+1} composed of ∇=2 bits u_i and u_{i+1}, where u_i is the i^{th} bit of the information bits U contained in the received rotated and modulated symbol $x_{r,q}$, the MAP decoder provides 2^{∇}=4 *a posteriori* probabilities with the full knowledge of the received symbol $x_{r,q}$ by the decoder. The hard decision is the corresponding value u_iu_{i+1} that maximizes the *a posteriori* probability. These probabilities can be expressed in terms of joint probabilities.

$$Pr(d_k = u_iu_{i+1}|x_{r,q}) = \frac{p(d_k = u_iu_{i+1}, x_{r,q})}{\sum_{k=0}^{2^{\nabla}-1} p(d_k = u_iu_{i+1}, x_{r,q})} \tag{2.2}$$

The trellis structure of the code allows to decompose the calculation of joint probabilities between past and future observations. This decomposition utilizes the forward recursion metric $\alpha_k(s)$ (the probability of a state of the trellis at instant k computed from past values), backward recursion metric $\beta_k(s)$ (the probability of a state of the trellis at instant k computed from future values), and a metric $\gamma_k(s', s)$ (the probability of a transition between two states $s'$of the trellis). Using these metrics the expression of 2.2 becomes:

$$p(d_k = u_iu_{i+1}|x_{r,q}) = \sum_{(s',s)/d_k=u_iu_{i+1}} \beta_{k+1}(s)\alpha(s')\gamma_k(s', s) \tag{2.3}$$

The forward $\alpha_{k+1}(s)$ and backward $\beta_k(s)$ recursion metrics are calculated as follows.

$$\alpha_{k+1}(s) = \sum_{s'=0}^{2^{\nu}-1} \alpha_k(s')\gamma_k(s', s), \quad \text{for } l = 0 \ldots N-1 \tag{2.4}$$

$$\beta_k(s) = \sum_{s'=0}^{2^{\nu}-1} \beta_{k+1}(s')\gamma_k(s, s'), \quad \text{for } i = N-1 \ldots 0 \tag{2.5}$$

where ν designates the number of the encoder memory elements. Thus the encoder state number is equal to 2^{ν}. In our case ν=3.

The initialization of these metrics depends on the knowledge of initial and final state of the trellis, e.g if the encoder starts at state s_0 then $\alpha_0(s_0)$ has value 1 while other $\alpha_0(s)$ will be 0. If the initial state is unknown then all states are initialized to same equiprobable value.

Similarly the branch metric $\gamma_k(s', s)$ can be expressed as:

$$\gamma_k(s', s) = p(x_{r,q}|s_{r,q}).\Pr^a(d_k = d_k(s', s)) \tag{2.6}$$

where $\Pr^a(d_k$=$d_k(s', s))$ designates the *a priori* probability corresponding to transition from s' to s. $\Pr^a(d_k$=$d_k(s', s))$ is equal to 0 if the transition does not exist in the trellis. Otherwise its value depends

upon the statistics of the source. For an equiprobable source, $\Pr^a(d_k{=}d_k(s',s)){=}\frac{1}{2^q}$. $p(x_{r,q}|s_{r,q})$ represents the channel transition probability of the received rotated symbol $x_{r,q}$ and the transmitted rotated symbol $s_{r,q}$. This probability can be expressed for BPSK modulation scheme and for a Gaussian channel as follows.

$$p(x_{r,q}|s_{r,q}) = \prod_{m=1}^{M} \left(\frac{1}{\sigma\sqrt{2\pi}} . e^{-\frac{\left(x^m_{r,q} - s^m_{r,q}\right)^2}{\sigma^2}} \right) \tag{2.7}$$

where $x^i_{r,q}$ and $s^i_{r,q}$ are the i^{th} bit of the received $x_{r,q}$ and transmitted $s_{r,q}$ modulated symbols respectively. σ^2 is the variance of the noise.

The extrinsic information generated by the decoder is computed in the same way as the *a posteriori* information (equation (2.2)) but with a modified branch metric:

$$Pr^{ex}(d_k = u_i u_{i+1}|x_{r,q}) = \frac{\sum_{(s',s)/d_k = u_i u_{i+1}} \beta_{k+1}(s)\alpha(s')\gamma^{ex}_k(s',s)}{\sum_{(s',s)} \beta_{k+1}(s)\alpha(s^{'})\gamma^{ex}_k(s',s)} \tag{2.8}$$

Hence the branch metric does not take into account the already available information of a symbol for which extrinsic information is being generated. For parallel convolutional turbo codes, systematic part should removed from the branch metric computation.

2.2.2 Max-Log-MAP Decoding Algorithm

Using input symbols and *a priori* extrinsic information, each SISO decoder computes *a posteriori* LLRs. The SISO decoder computes first the branch metrics γ. Then it computes the forward α_k and backward β_k metrics between two trellis states s and s'.

$$\alpha_k(s) = \max_{(s',s)}(\alpha_{k-1}(s^{'}) + \gamma_k(s^{'},s)) \tag{2.9}$$

$$\beta_k(s) = \max_{(s',s)}(\beta_{k+1}(s^{'}) + \gamma_{k+1}(s^{'},s)) \tag{2.10}$$

where

$$\gamma_k(s^{'},s) = \gamma^{Sys}_k(s^{'},s) + \gamma^{Parity}_k(s^{'},s) + \gamma^{Ext}_k(s^{'},s) \tag{2.11}$$

The soft output information $L^{apost}_{Dec}(d_k = u_i u_{i+1})$ and symbol-level extrinsic information $L^{ext}_{Dec}(d_k = u_i u_{i+1})$ of symbol k are then computed using equations (2.12) and (2.13). The extrinsic information, which is exchanged iteratively between the two SISO decoders, is obtained by subtracting the intrinsic information from $so(d_k = u_i u_{i+1})$.

$$L^{apost}_{Dec}(d_k) = \max_{(s',s)/d(s',s)=d_k} \Big(\alpha_{k-1}(s^{'}) + \gamma_k(s^{'},s) + \beta_k(s)\Big) \tag{2.12}$$

$$L^{ext}_{Dec}(d_k) = \max_{(s',s)/d(s',s)=d_k} \Big(\alpha_{k-1}(s^{'}) + \gamma^{Ext}_k(s^{'},s) + \beta_k(s)\Big) \tag{2.13}$$

$L_{Dec}^{ext}(d_k)$ can be multiplied by a constant scaling factor SF (typically equals to 0.75) for a modified Max-Log-MAP algorithm improving the resultant error rate performance.

Finally, in case of turbo demapping and only by one SISO decoder, the bit-level extrinsic information of systematic symbols $u_i u_{i+1}$ are computed using equations (2.14) and (2.15). Similar computations are done for parity symbols.

$$L_{Dem}^{apr}(u_i) = \max[z(d_k = 11), z(d_k = 10)] - \max[z(d_k = 01), z(d_k = 00)] \quad (2.14)$$

$$L_{Dem}^{apr}(u_{i+1}) = \max[z(d_k = 11), z(d_k = 01)] - \max[z(d_k = 10), z(d_k = 00)] \quad (2.15)$$

These expressions exhibit three main computation steps: (a) branch metrics computation referred by γ_k, (b) state metrics computation referred by (α_k and β_k), and (c) extrinsic information computation referred by L_{Dem}^{apr} and $L_{Dec}^{ext}(d_k)$.

2.2.3 Parallelism in Turbo Decoding

In turbo decoding with the Max-Log-MAP algorithm executing inside the decoder, the parallelism can be classified at three levels [36] [37]: (1) Metric level, (2) SISO decoder level, and (3) Turbo decoding level. The first (lowest) parallelism level concerns the elementary computations for LLR generation inside a SISO decoder. Parallelism between these SISO components, inside a turbo decoding process, belongs to the second parallelism level. The third (highest) parallelism level duplicates the whole turbo decoder hardware itself.

2.2.3.1 Metric Level Parallelism

The parallelism level of BCJR metrics computation addresses the parallelism available in the computation of all the metrics required to decode each received symbol within a BCJR-SISO decoder. This parallelism level exploits both the inherent parallelism of the trellis structure [38, 39], and the available parallelism in BCJR computations [38–40].

Parallelism of Trellis Transition: The first parallelism available in the Max-Log-MAP algorithm is the computation of the metrics associated to each transition of a trellis (Fig. 2.2). These metrics are γ, α, β, and the extrinsic information. Trellis-transition parallelism can easily be extracted from the trellis structure as the same operations are repeated for all transitions. The first metric (γ) calculation is completely parallelizable with a degree of parallelism naturally bounded by the number of transitions in the trellis. But in practice, the degree of parallelism associated with computing the branch metric is bounded by the number of possible binary combinations of input and parity bits. Thus, several transitions may have the same probability in a trellis. The other metrics α, β, and extrinsic computation can be parallelized with a bound of total number of transitions in a trellis. Furthermore this parallelism implies low area overhead as only the computational units have to be duplicated. In particular, no additional memories are required since all the parallelized operations are executed on the same trellis section, and in consequence on the same data.

Parallelism of BCJR Computations: A second metric parallelism can be orthogonally extracted from the BCJR algorithm through a parallel execution of the three BCJR computations (α, β, and extrinsic computation). The parallel execution of these computations was proposed with the original Forward-Backward scheme. In this scheme, the BCJR computation parallelism degree is equal to one in the forward part and two in the backward part.

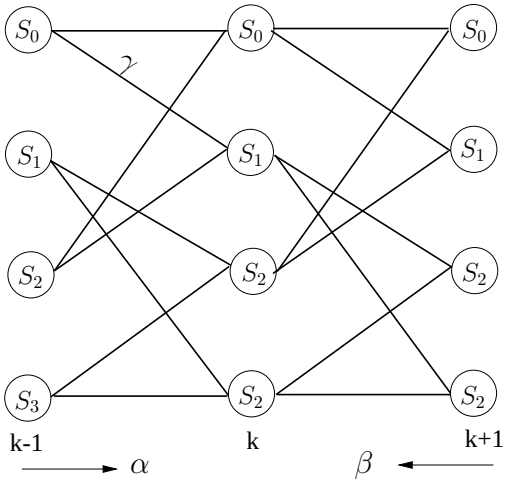


Figure 2.2: BCJR metrics associated with trellis transitions

To increase this parallelism degree, several schemes are proposed [40]. One of these schemes, the butterfly scheme, doubles the parallelism degree of the original scheme through the parallelism between the forward and backward recursion computations. This is performed without any memory increase and only the BCJR computation resources have to be duplicated. Thus, metric computation parallelism is area efficient but still limited in parallelism degree.

2.2.3.2 SISO Decoder Level Parallelism

The second level of parallelism concerns the SISO decoder level. It consists of the use of multiple SISO decoders, each executing the BCJR algorithm and processing a sub-block of the same frame in one of the two interleaving orders. At this level, parallelism can be applied either on sub-blocks and/or on component decoders.

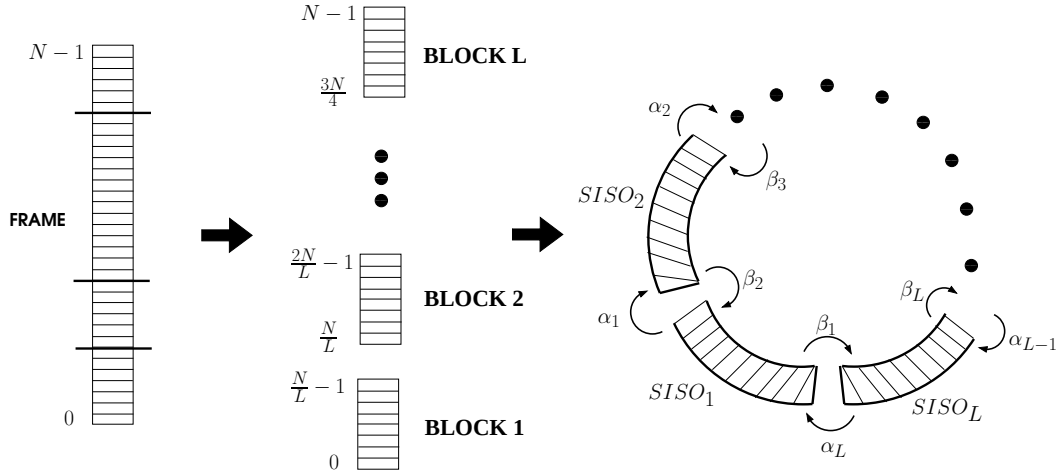


Figure 2.3: Sub-block parallelism with message passing for metric initialization, circular code.

Frame Sub-blocking: In sub-block parallelism, each frame is divided into L sub-blocks and then each sub-block is processed by a BCJR-SISO decoder using adequate initializations as shown in Fig.

2.3. Besides duplication of BCJR-SISO decoders, this parallelism imposes two other constraints:

- Interleaving has to be parallelized in order to scale proportionally the communication bandwidth. Due to the scramble property of interleaving, this parallelism can induce communication conflicts except for interleavers of emerging standards that are conflict-free for certain parallelism degrees. These conflicts force the communication structure to implement conflict management mechanisms and imply a long and variable communication time. This issue is generally addressed by minimizing interleaving delay with specific communication networks [41].
- BCJR-SISO decoders have to be initialized adequately either by acquisition or by message passing.

Acquisition method has two implications on implementation. First of all extra memory is required to store the overlapping windows when frame sub-blocking is used and secondly extra time will be required for performing acquisition. Other method, the message passing, which initializes a sub-block with recursion metrics computed during the previous iteration in the neighboring sub-blocks, needs not to store the recursion metric and time overhead is negligible. In [42] a detailed analysis of the parallelism efficiency of these two methods is presented which gives favor to the use of message passing technique.

Shuffled Turbo Decoding: The basic idea of shuffled decoding technique [43] is to execute all component decoders in parallel and to exchange extrinsic information as soon as it is created, so that component decoders use more reliable *a priori* information. Thus the shuffled decoding technique performs decoding (computation time) and interleaving (communication time) fully concurrently while serial decoding implies waiting for the update of all extrinsic information before starting the next half iteration (Fig. 2.4). Thus, by doubling the number of BCJR-SISO decoders, component-decoder parallelism halves the iteration period in comparison with originally proposed serial turbo decoding.

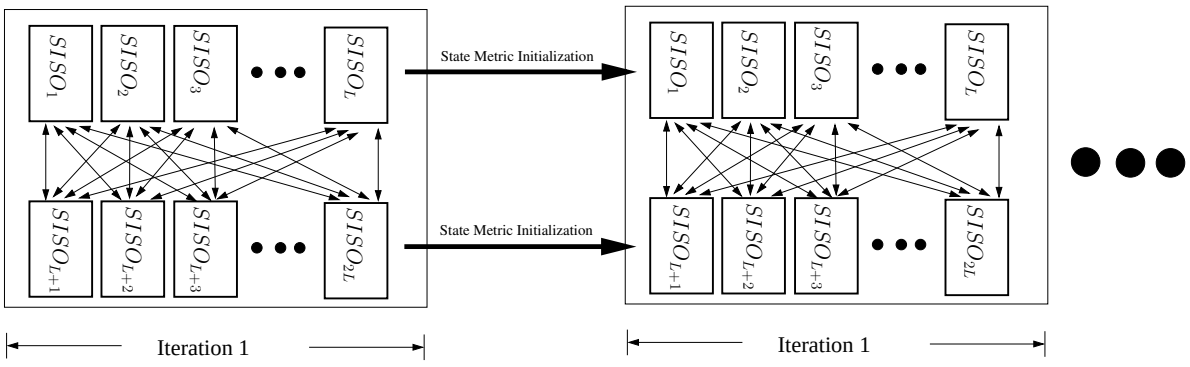


Figure 2.4: Shuffled turbo decoding.

Nevertheless,to preserve error-rate performance with shuffled turbo decoding, an overhead of iteration between 5 and 50 percent is required depending on the BCJR computation scheme, on the degree of sub-block parallelism, on propagation time, and on interleaving rules [42].

2.2.3.3 Parallelism of Turbo Decoder

The highest level of parallelism simply duplicates whole turbo decoders to process iterations and/or frames in parallel. Iteration parallelism occurs in a pipelined fashion with a maximum pipeline depth

equal to the iteration number, whereas frame parallelism presents no limitation in parallelism degree. Nevertheless, turbo-decoder level parallelism is too area-expensive (all memories and computation resources are duplicated) and presents no gain in frame decoding latency.

2.3 SISO Turbo Decoding Schemes

The Max-Log-MAP decoding algorithm presented in subsection 2.2.2 computes extrinsic information by means of two recursive operations: forward (α) and backward (β) state metric computations. Depending on the way the forward and backward recursions are executed, several schemes can be applied. In this following, the SISO decoder scheme or simply scheme refers to the organization of the BCJR algorithm computations inside the SISO decoder.

Fig. 2.5 depicts three different schemes when the decoding process is applied over a block of N information symbols. The horizontal axis represents the time and the vertical axis the current information symbol processed by the SISO decoder. Continuous lines symbolize α or β computations, and dashed lines state metric along with extrinsic information computations. The gray area represents the time interval during which α or β values are kept in a memory.

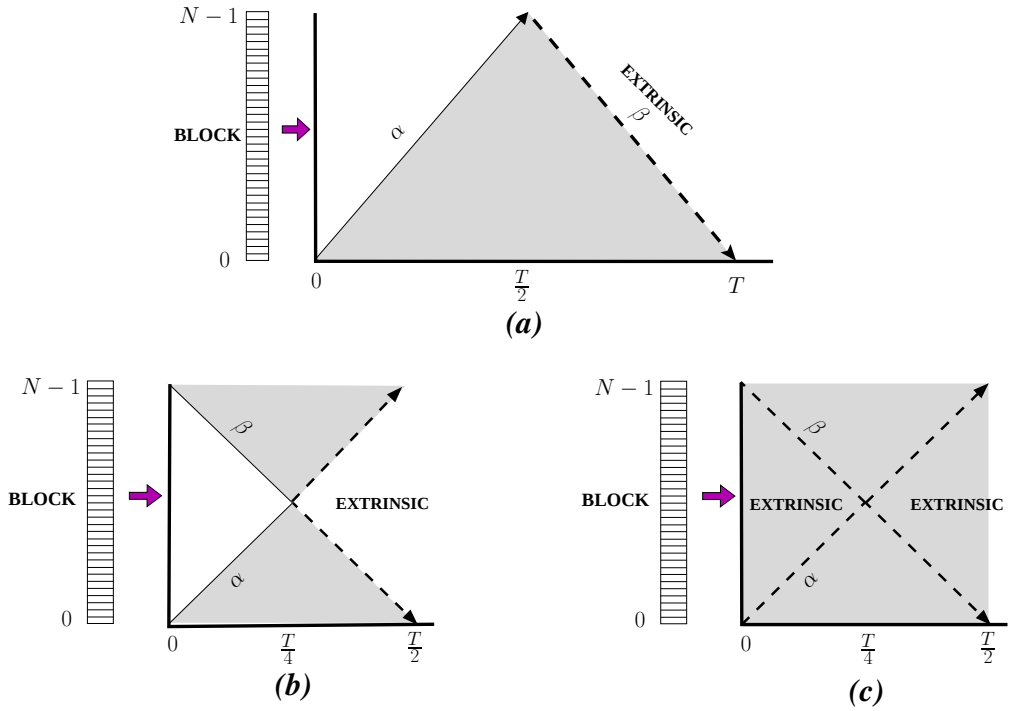


Figure 2.5: SISO decoder schemes: (a) Forward-Backward (b) Butterfly (c) Butterfly-Replica.

Fig. 2.5(a) depicts the classical scheme called Forward-Backward (*F-B*). In this scheme the Max-Log-MAP algorithm begins by computing recursively, starting from the beginning to the end of the block, α state metrics. These values will be memorized for later use. Once all α values are computed, β state metric computation starts. In parallel, thanks to the α values previously computed, extrinsic values are calculated. Let us denote by T the time needed to generate all the extrinsic values of the block when the scheme *F-B* is used. This time is assumed to be proportional to N. It can be halved by using the Butterfly (*B*) scheme (Fig. 2.5(b)), at the cost of doubling the hardware resources required to compute α, β and the extrinsic information. On the other hand, the Butterfly-Replica (*B-R*) Scheme (Fig. 2.5(c)) , originally proposed in [42], has the same decoding duration than *B*.

However, it generates extrinsic values continuously all along the block decoding process. To this end, in comparison with *B*, the state metric values generated after the time $T/4$ have to be stored in order to be used in the next iteration.

The *B-R* scheme was originally proposed to improve the convergence of the shuffled turbo decoding by an intensive exchange of the extrinsic information values. Two decoder extrinsic values are generated for each information symbol per iteration. Note that the use of *B-R* scheme in no-shuffled turbo decoding will not provide any improvement with respect to the *B* scheme since the extrinsic values are only read at the end of each SISO decoding process. Therefore, extrinsic values generated for $T < T/4$ will not be considered.

The height of the gray area in figure 2.5 represents the size of the memory needed to store α and β state metrics. The three considered schemes require the same memory size equal to N. In order to reduce the size of this memory and decoding delay, sliding window technique was proposed [44]. In this technique the decoding algorithm is executed over a length of a window, smaller than the block length. The size of the state metric memory is then reduced to the length of the window. Adjacent windows are processed sequentially, hence the memory size is significantly reduced. Nevertheless the window size should not be taken arbitrarily small, otherwise the BER performances will degrade. On the other hand, sliding window technique is not recommended with *B-R*. In this case, state metric values have to be kept to the next iteration for all block symbols and the memory used in a window cannot be reused over successive windows in the block. Hence, *B* and *B-R* schemes will not have the same memory area.

For high throughput receivers, the sub-block technique with large number of SISO decoders is necessary. Hence, the size of the sub-block becomes comparable to the minimum window size required to avoid significant performance degradation. In this case, the memories assigned for *B* and *B-R* schemes have exactly the same sizes.

Extensive simulations have been made, showing a minimum sub-block size of 64 symbols to avoid important performance degradation. On the other hand, the maximum sub-block size is assumed to be 128 symbols. For the rest of this section, we will consider only the *B* and *B-R* schemes which provide higher parallelism degree.

2.4 Turbo Decoding Schedulings

Turbo Decoding Scheduling refers to the organization of the BCJR algorithm computations in natural domain according to the BCJR algorithm computations in interleaved domain. It determines when the activity of the SISO decoder in natural or interleaved domain should start or end based on the resulted BER performance results, throughput and latency. Depending on the turbo decoder processing mode, many schedulings can be proposed.

2.4.1 Classical Turbo Decoding Schedulings

The decoding approach proposed in [15], and shown in Fig. 2.6a, operates in serial mode, i.e., the component decoders $\mathtt{DEC}_1$ and $\mathtt{DEC}_2$ take turns generating the extrinsic values of the estimated information symbols, and each component decoder uses the extrinsic messages delivered by the last component decoder as the *a priori* values of the information symbols. The disadvantage of this scheme is high decoding delay and low throughput. In the parallel turbo decoding algorithm [45] (Fig. 2.6b), all component decoders operate in parallel at any given time. After each iteration, the component

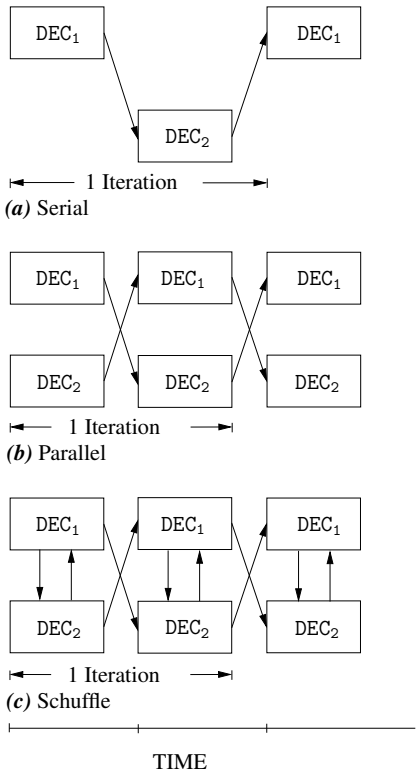


Figure 2.6: Classical turbo decoding schedulings: (a) Serial (b) Parallel (c) Shuffled

decoder delivers extrinsic messages to the other component decoder which use these messages as *a priori* values at the next iteration.

Although the parallel turbo decoding overcomes the drawback of high decoding delay of serial decoding, the extrinsic messages are not taken advantage of as soon as they are available, because the extrinsic messages are delivered to component decoders only after each iteration is completed. The aim of the shuffled turbo decoding is to use the more reliable extrinsic messages at each time as shown in Fig. 2.6c. In shuffled turbo decoding, the two component decoders $\texttt{DEC}_1$ and $\texttt{DEC}_2$ operate simultaneously as in the parallel turbo decoding scheme, but the scheme of updating and delivering messages is different. We assume that the two component decoders deliver extrinsic messages synchronously,

2.4.2 Shuffled Turbo Decoding scheduling with Overlapping

In this subsection, we introduce the concept of overlapping in the processing of the shuffled turbo decoding scheduling.

The *No Overlapping* scheme of Fig. 2.7a corresponds to the classical shuffled scheduling of Fig. 2.6c. The two component decoders $\texttt{DEC}_1$ and $\texttt{DEC}_2$ begin processing at the same time without any delay (δ=0). Introducing a normalized delay $\delta=\frac{delay}{N_{CSymb}}$ between the processing of $\texttt{DEC}_1$ and $\texttt{DEC}_2$ corresponds to Fig. 2.7b and 2.7c. $delay$ is expressed in terms of number of coded symbols, $0 \leq delay \leq N_{CSymb}$. Hence, $0 \leq \delta \leq 1$.

The *Overlapping 1* scheme of consecutive iterations is achieved by starting the next iteration before the end of the current iteration, as depicted in Fig. 2.7b. On the other hand, the *Overlapping 2* technique starts the processing of the next iteration once the current iteration is totally completed.

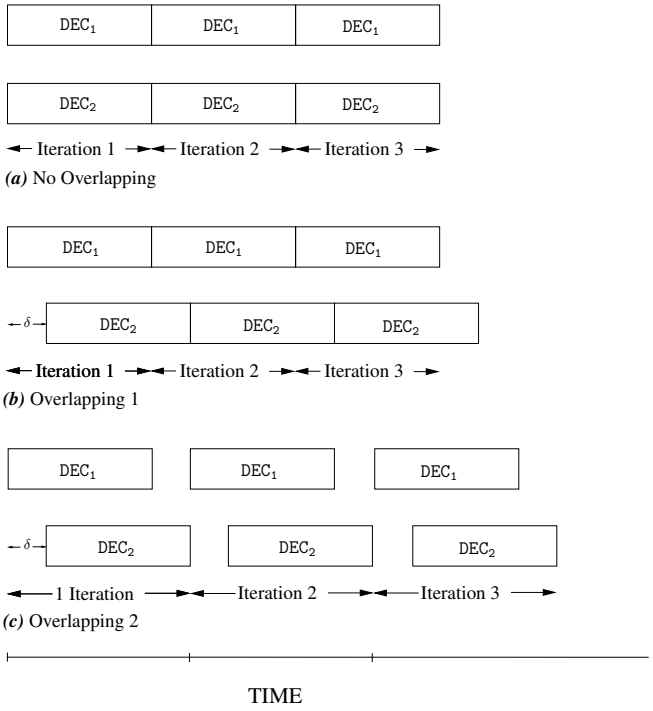


Figure 2.7: Shuffled turbo decoding scheduling with overlapping: (a) No Overlapping (b) Overlapping 1 (c) Overlapping 2

Hence, the iteration duration of *Overlapping 2* is superior to the duration of *Overlapping 1* by δ. If δ=1, the *Overlapping 2* scheme of the shuffled mode will process as in serial mode (Fig. 2.6a).

2.4.2.1 *Overlapping 1* scheme

In order to evaluate the impact of δ on the *Overlapping 1* scheme, we plot Fig. 2.8. This figure shows the BER performance at the output of $\texttt{DEC}_2$ in function of iterations for the TBICM-SSD receiver processing in shuffled turbo decoding with *Overlapping 1* scheme for different values of δ. One system configuration, QAM16 and R_c=$\frac{1}{2}$, is considered. The butterfly scheme is used.

As we can see form Fig. 2.8, the red curve corresponds to the classical shuffle mode (δ=0). The other colored curves correspond to $\delta \neq 0$. The first value in the legend, after the value of δ, corresponds to the scaling factor of $\texttt{DEC}_1$. Meanwhile, the second value corresponds to the scaling factor of $\texttt{DEC}_2$.

For the first iteration, simulations show an improved BER performance values at the output of $\texttt{DEC}_2$ when considering higher δ values. This is due to the fact that $\texttt{DEC}_2$ is beginning the shuffled decoding process in the presence of already *a priori* information generated by $\texttt{DEC}_1$ during the time delay δ. After a specific number of iterations, simulations show that the curve corresponding to δ=0 outperform the other curves when considering the same scaling factor for $\texttt{DEC}_1$ and $\texttt{DEC}_2$ (equal to 0.75 for example). Taking the case of δ=1, 12 iterations are required to achieve BER=$4,5.10^{-5}$. Meanwhile, 11 iterations are required for δ=0 to achieve a lower BER value.

In order to improve the performances for $\delta \neq 0$, we propose to take different scaling factor values for $\texttt{DEC}_1$ and $\texttt{DEC}_2$. In fact, the scaling factor for $\texttt{DEC}_1$ should be considered smaller than for $\texttt{DEC}_2$ since the extrinsic information at this latter are more reliable at the same iteration. Many simulations have been launched to search for the modified scaling factor while keeping the second factor equals to 0.75 in order to make a good comparison with the other curves.

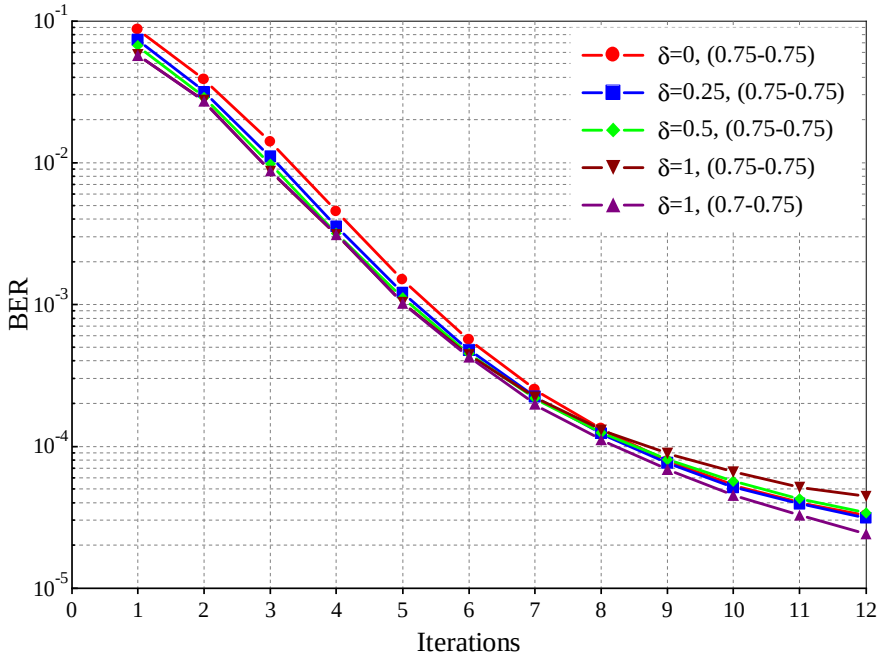


Figure 2.8: BER performance at the output of $\mathtt{DEC}_2$ in function of iterations for TBICM-SSD processing in shuffled turbo decoding with butterfly scheme and *Overlapping 1* scheme for different values of δ and scaling factors (2.2.2) for the transmission of 1536 information bits frame over Rayleigh fading channel. QAM16 modulation scheme, R_c=$\frac{1}{2}$ and E_b/N_0=6.25 dB are considered.

The modified scaling factor value which optimize the BER performance at the output of $\mathtt{DEC}_2$ is 0.7 for $\mathtt{DEC}_1$ while keeping unchanged the correspondent value for $\mathtt{DEC}_2$ equals to 0.75.

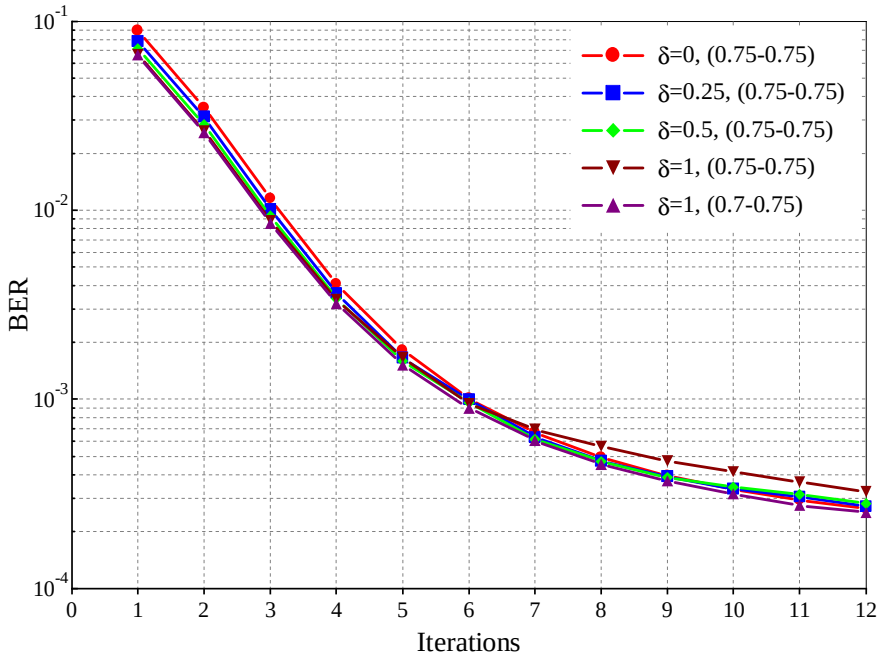


Figure 2.9: BER performance at the output of $\mathtt{DEC}_2$ in function of iterations for TBICM-SSD processing in shuffled turbo decoding with butterfly-replica scheme and *Overlapping 1* scheme for different values of δ and scaling factors (2.2.2) for the transmission of 1536 information bits frame over Rayleigh fading channel. QAM16 modulation scheme, R_c=$\frac{1}{2}$ and E_b/N_0=6 dB are considered.

Considering the modified scaling factor and δ=1 (which provides the best performances at the first iteration), the correspondent BER curve is plotted in purple color in Fig. 2.8. For BER=$4,5.10^{-5}$, results show a need of 10 iterations (purple curve) when applying the modified scaling factor instead of 12 iterations (brown curve). Moreover, for identical number of iterations, the proposed technique (δ=1 and the modified scaling factor) provides better BER performance than for the classical scheme (δ=0) at the expense of an additional latency of one iteration.

The extension of this analysis to the butterfly-replica scheme gives same results as shown in Fig. 2.9.

These improvements can be considered not significant enough for low and medium BER values. Hence, this scheme for shuffled turbo decoding will not be considered of the rest of this manuscript.

2.4.2.2 *Overlapping 2* scheme

Now, we will consider the *Overlapping 2* scheme. Fig. 2.10 illustrates the BER performance at the output of $\mathtt{DEC}_2$ in function of iterations for the TBICM-SSD receiver processing in shuffled turbo decoding with *Overlapping 2* scheme for different values of δ. One system configuration, QAM16 and R_c=$\frac{1}{2}$, is considered. The butterfly schemes is applied.

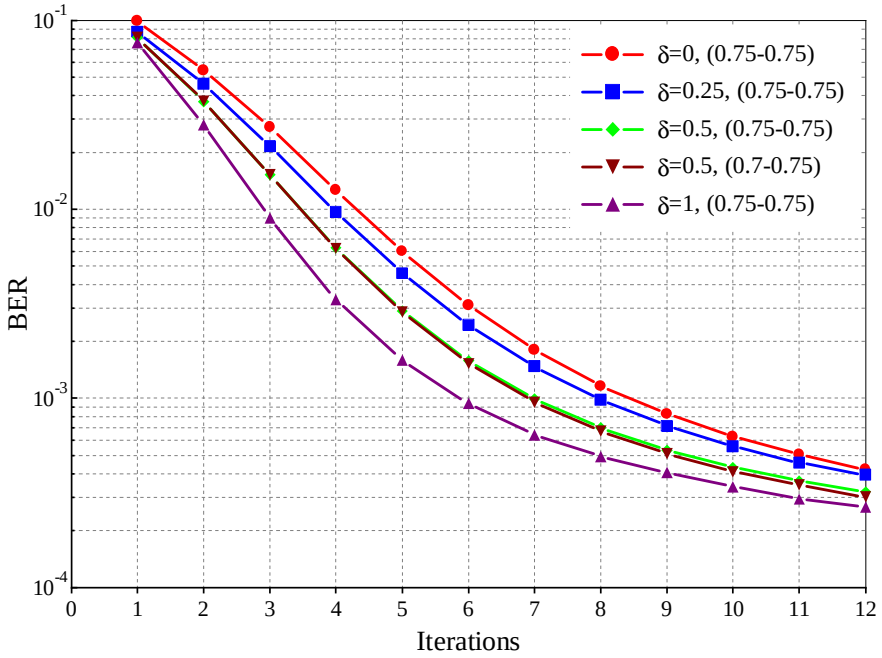


Figure 2.10: BER performance at the output of $\mathtt{DEC}_2$ in function of iterations for TBICM-SSD processing in shuffled turbo decoding with butterfly scheme and *Overlapping 2* scheme for different values of δ and scaling factors (2.2.2) for the transmission of 1536 information bits frame over Rayleigh fading channel. QAM16 modulation scheme, R_c=$\frac{1}{2}$ and E_b/N_0=6 dB are considered.

The red curve of Fig. 2.10 corresponds to the classical shuffle mode (δ=0). The other colored curves correspond to $\delta \neq 0$.

Fig. 2.10 performance simulations show improved BER performance values at the output of $\mathtt{DEC}_2$ at each iteration when considering higher δ values. This improvement is shown at the expense of an additional delay δ for each decoding iteration.

In fact, it is known that the serial decoding (δ=1) provides better performances at each iteration in comparison to the shuffled decoding. Thus, increasing δ from 0 to 1 will improve the performances

at each iteration. Using this technique, a scalable tradeoff scheme between the serial and shuffled decoding can be achieved by changing the value of δ.

In order to improve the performances for $\delta \neq \{0,1\}$, we have tried different scaling factor values for $\texttt{DEC}_1$ and $\texttt{DEC}_2$. Results shows identical values as in subsection 2.4.2.1: 0.7 for $\texttt{DEC}_1$ and 0.75 for $\texttt{DEC}_2$.

Taking the example of δ=0.5 and the modified scaling factor, the correspondent BER curve (brown color) is plotted in Fig. 2.10. Results show almost similar performances than the case with identical scaling factor values (green curve). Hence, the modified scaling factor technique does not provide any advantage for the *Overlapping 2* scheme.

Similar analysis has been conducted for the butterfly-replica case. The results where the same.

2.5 Summary

In this chapter, we have presented the SISO algorithm related to convolutional turbo decoding. Simplified expressions of the considered algorithm, suitable for hardware implementations, were also provided. Parallelism techniques addressing the issues of latency and low throughput associated with turbo decoding were also discussed.

Moreover, two proposals for shuffled turbo decoding scheduling were presented and analyzed in this chapter for butterfly and butterfly-replica schemes. The idea consists of investigating the introduction a time delay (δ) between the processing of the natural and interleaved domains constituent component decoders. The first proposal has shown a slight improvement in comparison to the original shuffled decoding. Meanwhile, the second proposal enables to realize any compromise between the serial and the shuffled turbo decoding schedulings in terms of error correction performance at each iteration.

CHAPTER

3 Optimized Turbo Demodulation with Turbo Decoding: Algorithms, Schedulings, and Complexity Estimation

WHILE the previous chapter has addressed the algorithmic and parallelism aspects of turbo decoding, this chapter investigates the iterative demodulation (ID) receiver TBICM-ID-SSD applying additional iterative feedback loop to the demapper.

Flexible and iterative baseband receivers with advanced channel codes like turbo codes are widely adopted nowadays, ensuring promising error rate performances. Extension of this principle with feedback loop to the demapping function has proven to provide substantial error performance gain at the cost of increased complexity. However, this complexity overhead constitutes commonly an obstacle for its consideration in real implementations.

In this chapter, after a brief review of state-of-the-art efforts in this domain, we introduce the SISO demapping algorithms and the different parallelism techniques for turbo demodulation. Then, Section 3.3 analyzes the convergence speed of the combined two iterative processes (turbo demodulation and turbo decoding) in order to determine the exact required number of iterations at each level. EXtrinsic Information Transfer (EXIT) charts are used for a thorough analysis at different modulation orders and code rates. An original iteration scheduling is proposed, in Section 3.4, which allows reducing two demapping iterations with reasonable performance loss of less than 0.15 dB. Normalizing and analyzing the computational and memory access complexity, which directly impact latency and power consumption, demonstrate the considerable gains of the proposed scheduling and the promising contributions of the proposed analysis. In fact, the analyzed complexity (number of arithmetic operations and read/write memory accesses) is independent from the architecture mode (serial or parallel) and remains valid for both of them. In order to reduce latency and increase throughput, parallel architectures can be implemented with different parallelisms degrees and techniques (operation-level or data-level through sub-blocking). However, all these architecture alternatives should execute the same number of operations (serially or concurrently) to process a received frame. For that reason, and for generalization and comparison fairness, the normalized technique was applied in serial mode. Furthermore, other scheduling ideas for TBICM-ID-SSD are explored and presented in Section 3.5.

Section 3.6 proposes a complexity adaptive iterative receiver performing TBICM-ID-SSD, illustrating the opposite of what is commonly assumed. Targeting identical error rate, results show that for certain system configurations, the TBICM-ID-SSD mode presents lower complexity than the TBICM-SSD. This original result is obtained when considering the equivalent number of iterations through detailed analysis of the corresponding computational and memory access complexity. The analysis is conducted for different parameters in terms of modulation orders and code rates and independently from the architecture for a fair comparison.

The last two sections of this chapter presents two further contributions which have been proposed as a joint work with two other PhD students. The first one, a joint contribution with Vianney Lapotre, proposes an efficient sizing of heterogeneous multiprocessor flexible iterative receiver implementing turbo demapping with turbo decoding. In fact, for a given communication requirement many architecture alternatives exist and selecting the right one at design-time and at run-time is an essential issue. The proposed approach defines the mathematical expressions which exhibit the number of heterogeneous cores and their features. The second contribution, a joint contribution with Oscar Sanchez, proposes to extend the use of the butterfly-replica scheme, originally proposed for shuffled turbo decoding, to full shuffled receiver implementing iterative demapping with turbo decoding. Simulation results show that applying this scheme in the turbo decoder reduces the overall number of iterations by at least one iteration in the waterfall region with respect to the butterfly scheme. In order to evaluate the impact on complexity and throughput, a detailed analysis is provided for different system configurations.

3.1 State of the Art

The BICM principle [46], first introduced in **1997**, represents the **state-of-the-art in coded modulations over fading channels**. The Bit-Interleaved Coded Modulation with Iterative Demapping (**BICM-ID**) scheme proposed in [47] is based on BICM with additional soft feedback from the SISO **convolutional decoder** to the constellation demapper. In this context, several techniques and configurations have been explored. In [17], the authors investigated different mapping techniques suited for BICM-ID and QAM16 constellations. They proposed several mapping schemes providing significant coding gains.

In [28], the convolutional code classically used in BICM-ID schemes was replaced by a **turbo code** in 1999. Only a small gain of 0.1 dB was observed. This result makes BICM-ID with turbo-like coding solutions (**TBICM-ID**) unsatisfactory with respect to the added decoding complexity. On the other hand, Signal Space Diversity (**SSD**) technique, which consists of a rotation of the constellation followed by a signal space component interleaving, has been recently proposed [7, 8]. It increases the diversity order of a communication system without using extra bandwidth.

Since 2005, several related contributions have been proposed by Telecom Bretagne. **Combining SSD technique with TBICM-ID** at the receiver side has shown excellent error rate performance results particularly in severe channel conditions (erasure, multi-path, real fading models) [19, 20]. It has shown additional error correction at the receiver side in an iterative processing scenario. In [19], the authors have proposed to combine SSD with TBICM-ID at the receiver and focuses mainly on error rate performance. Using EXIT charts, it has been proposed implicitly to apply one turbo decoding iteration for each demapping iteration without an explicit analysis of the convergence speed. In addition, the complexity aspects was not discussed in this work.

The authors in [48, 49] have conducted a **parallelism study of turbo demodulation combined with turbo decoding**. Speed gains and parallelism efficiency obtained with various parallelism techniques in a turbo demodulation process have been evaluated. The iterations scheduling adopted here applies one turbo decoding iteration for each demodulation iteration. In addition, a flexible hardware architecture and FPGA prototype for SISO demapper have been proposed [50, 51] based on the ASIP concept (Application-Specific Instruction-set Processor). The flexibility of the designed DemASIP allows its reuse for BPSK to QAM256 constellation (with/without SSD) and any mapping scheme.

In [20], the authors have proposed to **use the SSD technique in turbo demodulation associated with LDPC** in order to increase the diversity order of coded modulations over fading channels. The obtained results were behind the adoption of this system in the DVB-T2 standard (using LDPC channel code). Targeting DVB-T2 standard, hardware implementation aspects have been considered in [21]. A demodulation based on the decomposition of the constellation into two-dimensional subregions in signal space associated to an algorithmic simplification were presented. Several optimizations techniques for efficient design and FPGA hardware prototyping of the considered architecture where also realized.

Turbo demodulation associated with LDPC decoding was also investigated by a research group from RWTH Aachen University in [52] in the context of the DVB-S2 standard. The authors have shown an improvement of about 0.3 dB with a slight increase of the receiver complexity.

In fact, most of the existing works have considered the optimization of individual SISO components without deep investigation of potential optimisation techniques from a system-level point of view. The application of the iterative demapping in wireless receivers using advanced iterative channel decoding leads to further latency problems, more power consumption, and more complexity caused by feedback inner and outer the decoder. Besides extrinsic information exchange inside the iterative channel decoder, additional extrinsic information is fed back as *a priori* information used by

the demapper to improve the symbol to bit conversion. Hence, the number of iterations to be run at each level should be determined accurately as it impacts significantly, besides error rate performance, latency, power consumption, and complexity.

3.2 SISO Demapping Algorithms

In this section, the basic receiver model considered in the previous chapter has been further extended by including a feedback from the decoder to the demapper as shown in Fig. 3.1. The considered receiver model can apply turbo demodulation scheme in combination with turbo decoding.

The channel model considered is a frequency non-selective memoryless channel with erasure probability. The received discrete time baseband rotated complex signal $\mathbf{X'_r} = \{x'_{r,0}, x'_{r,1}, \ldots x'_{r,q}, \ldots\}$ can be written as:

$$\begin{aligned} x'_{r,q} &= h_q.\rho_q.s'_{r,q} + n_q \\ &= h'_q.s'_{r,q} + n_q \end{aligned} \tag{3.1}$$

where r and q subscripts designate respectively the rotation term and the index of the symbol. h_q is the Rayleigh fast-fading coefficient, ρ_q is the erasure coefficient taking value 0 with a probability P_ρ and value 1 with a probability of $1 - P_\rho$. n_q is a complex additive white Gaussian noise with spectral density $N_0/2$ in each component axe, and h'_q is the channel attenuation. Note that, at the receiver side, the transmitted energy has to be normalized by a $\sqrt{1 - P_\rho}$ factor in order to cope with the loss of transmitted power due to erasure events.

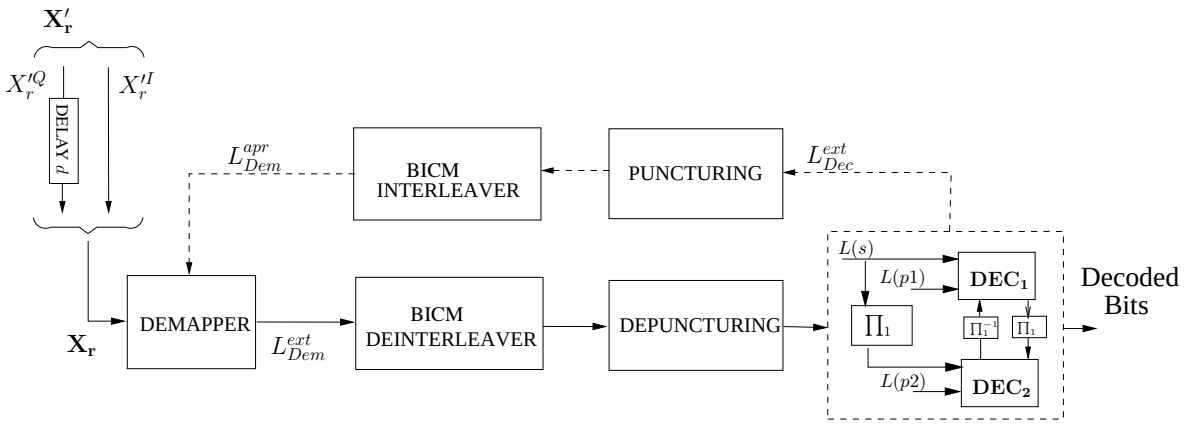


Figure 3.1: Receiver model with TBICM-ID-SSD.

At this side, the complex received symbols $x'_{r,q}$ have their Q-components re-shifted resulting in $x_{r,q}$. A probability $P(c_{p,q} = l|x_{r,q})$ with $l \in \{0, 1\}$ or an extrinsic log-likelihood ratio $L^{ext}_{Dem}(c_{p,q}/x_{r,q})$ is calculated for each coded bit $c_{p,q}$ corresponding to the p^{th} bit of the received rotated and modulated symbol $x_{r,q}$. After de-interleaving, de-puncturing and turbo decoding, extrinsic information from the turbo decoder $L^{ext}_{Dec}(c_{p,q})$ is punctured, passed through the BICM interleaver, and fed back as *a priori* information to the demapper in a turbo demodulation scheme.

3.2.1 MAP Demapping Algorithm

The demapper produces probabilities on coded bits $c_{p,q}$. At modulated symbol q, the probability of error on bit $c_{p,q}$ noted $P(c_{p,q} = l|x_{r,q})$ with $l \in \{0, 1\}$ is expressed as follows [3]:

$$P(c_{p,q} = l|x_{r,q}) = \sum_{s_{r,j} \in \mathcal{X}_{r,l}^{p}} P(s_{r,q}|x_{r,q}) = \sum_{s_{r,j} \in \mathcal{X}_{r,l}^{p}} P(x_{r,q}|s_{r,q}).P(s_{r,q}) \tag{3.2}$$

where $\mathcal{X}_{r,l}^{p}$, are the symbol sets of the constellation for which symbols have their p^{th} bit equals to l. $P(s_{r,q})$ designates the *a priori* probability of $s_{r,q}$. In the presence of equiprobable source the $P(s_{r,q}) = 1$.

$P(x_{r,q}|s_{r,q})$ and $P(s_{r,q})$ can be expressed as:

$$P(x_{r,q}|s_{r,q}) = \frac{1}{\sigma\sqrt{\pi}} e^{-\frac{h_q'^2}{\sigma^2}|x_{r,q}^I - s_{r,j}^I|^2 + \frac{h_{q-1}'^2}{\sigma^2}|x_{r,q}^Q - s_{r,j}^Q|^2} \tag{3.3}$$

$$P(s_{r,q}) = \prod_{i=0, i\neq p}^{M-1} P(c_{i,q}) \tag{3.4}$$

where M is the number of bits per modulated symbol. $P(c_{i,q})$ is the probability of the i^{th} bit of constellation symbol $s_{r,q}$ computed through demapper *a priori* information.

3.2.2 Max-Log-MAP Demapping Algorithm

In practice, due to its complexity, the MAP algorithm is not implemented in its probabilistic form but rather used in logarithmic domain to simplify exponential operations and transform multiplications into additions. The extrinsic information $L_{Dem}^{ext}(c_{p,q}/x_{r,q})$ is calculated as the difference between the soft output *a posteriori* $L_{Dem}(c_{p,q}/x_{r,q})$ and the soft input *a priori* $L_{Dem}^{apr}(c_{p,q})$ at the demapper side. It was originally computed in [53] and given by the expression below:

$$\begin{aligned} L_{Dem}^{ext}(c_{p,q}/x_{r,q}) &= L_{Dem}(c_{p,q}/x_{r,q}) - L_{Dem}^{apr}(c_{p,q}) \\ &= \log\left(\frac{Z_1}{Z_2}\right) \end{aligned} \tag{3.5}$$

$Z_{l(l=0,1)}$ can be expressed as:

$$Z_{l(l=0,1)} = \sum_{s_{r,j} \in \mathcal{X}_{r,l}^{p}} e^{-A_q} . \prod_{i=0, i\neq p}^{M-1} P(c_{i,q}) \tag{3.6}$$

A_q is computed as follows.

$$A_q = \frac{h_q'^2}{\sigma^2}|x_{r,q}^I - s_{r,j}^I|^2 + \frac{h_{q-1}'^2}{\sigma^2}|x_{r,q}^Q - s_{r,j}^Q|^2 \tag{3.7}$$

Applying the Max-Log-MAP approximation, equation (3.5) becomes [53]:

$$L_{Dem}^{ext}(c_{p,q}/x_{r,q}) = \min_{s_{r,j} \in \mathcal{X}_{r,0}^{p}} (A_q - B_{p,q}) - \min_{s_{r,j} \in \mathcal{X}_{r,1}^{p}} (A_q - B_{p,q}) \tag{3.8}$$

where $B_{p,q}$ is computed as follows.

$$B_{p,q} = \left(\sum_{i=0, c_{i,q}=1}^{M-1} L_{Dem}^{apr}(c_{i,q}) \right) - L_{Dem}^{apr}(c_{p,q}) \tag{3.9}$$

The above demapping equations are valid for both channel models (with or without erasures) through the use of h'_q coefficient (equation (3.1)).

These simplified expressions exhibit three main computation steps: (a) Euclidean distance computation referred by A_q, (b) *a priori* adder operation referred by $B_{p,q}$, and (c) minimum finder operation referred by the min operations of equation (3.8).

More reductions can be achieved for medium and high constellation sizes (as for QAM64, QAM256) by applying the sub-partitioning technique of the constellation as presented in [21]. This technique reduces the search of closest constellation point complexity order from 2^M to $(2^{\frac{M-2}{2}}+1)^2$.

In the context of Gray mapping, no constellation rotation, and the absence of *a priori* information, equation (3.8) can be simplified as follows:

$$L_{Dem}^{ext}(c_{p,q}/x_q) = \frac{1}{\sigma^2}[|x_q^o - h'_q.\tilde{s}_{p,0}^o|^2 + |x_q^o - h'_{q-1}.\tilde{s}_{p,1}^o|^2] \tag{3.10}$$

where $o \in \{I, Q\}$. $\tilde{s}_{p,0}^o$ and $\tilde{s}_{p,1}^o$ represent the symbols that give the two *min* operations of equation (3.8). Hence, taking the values of $\tilde{s}_{p,0}^o$ and $\tilde{s}_{p,1}^o$ for each region of the constellation for each modulation scheme can lead to simple equations for symbol ($x_{r,q}$) to LLR ($L_{Dem}^{ext}(c_{p,q}/x_{r,q})$) conversion.

3.2.3 Parallelism in Turbo Demodulation

In a turbo demodulation receiver, the parallelism can be categorized into three levels [49]:

- Metric Level Parallelism
- Demapper Component Level Parallelism
- Turbo Demodulation Level Parallelism.

3.2.3.1 Demapping Metric Level Parallelism

The metric level parallelism concerns the concurrent computations of the Euclidean distances A_q (equation (3.7)), the sum of the *a priori* information $B_{p,q}$ (equation (3.9)) and the two minimum operations of equation (3.8).

In a constellation with M bits per modulated symbol, one needs $M2^M$ Euclidean distances and $M(M-1)$ $a\ priori$ addition operations which are then fed to $2M$ minimum operations each having to process 2^{M-1} Euclidean distances. This illustrates the complexity and the maximum parallelism degree at this level, which varie significantly with M.

3.2.3.2 Demapper Component Level Parallelism

Same as in turbo decoding, there are two categories at this level: sub-block parallelism and shuffled turbo demodulation.

Frame Sub-blocking: At this level, the demapping process is independent from the modulated frame length. Hence, there are no issue of SISO sub-block initialization compared to turbo decoding. Therefore, linear increase in throughput can be achieved by using multiple SISO demappers processing in parallel.

Shuffled Turbo Demodulation: This type of parallelism is inherited from the concept of shuffled turbo decoding to execute both the decoding and demodulation tasks concurrently. In this scheme, all SISO demappers and SISO decoders components are executed simultaneously. Once the demapper components receive the input data, demapping is performed for the first time without *a priori* information to fill the channel input memories of the decoder components. After that, both demapping and decoding processes run in a shuffled scheme exchanging extrinsic information as soon as created.

3.2.3.3 Turbo Demodulation Level Parallelism

The highest level of parallelism duplicates the whole turbo demodulator to process iterations and/or frames in parallel.

3.3 Turbo Demodulation with Turbo Decoding Convergence Speed Analysis

This section illustrates the impact of the constellation rotation, the effects of bits-to-symbol allocation scheme, and the effects of the Max-Log-MAP demapping algorithm on the convergence speed of the iterative receiver. Convergence speed designates the rapidity of the convergence of the iterative process. Both TBICM-SSD and TBICM-ID-SSD system configurations are considered. For TBICM-ID-SSD, the impact of the number of turbo demapping and decoding iterations is analyzed. In this context, two types of iterations exist:

1. Iterations inside the decoder (Turbo decoding)
2. Iterations outside the decoder (Turbo demodulation, additional feedback from decoder to the demapper)

EXIT charts will be plotted in order to analyze the convergence speed and to ensure optimized number of iterations inside and outside the turbo decoder. In fact, when investigating error correction performance of an iterative process, typically three distinct regions can be identified as shown in Fig. 3.2:

- At very low SNR, the error performance is poor and is certainly not suitable for most communication systems.
- At medium SNR, the error performance curve improves steeply, providing very low error rates at moderate SNR. The region associated with the start and end of the steep error performance curve is known as the waterfall region. In this region, the performance is mainly determined by the convergence behavior of the turbo decoder. Generally, the longer the interleaver, the better the convergence and the better (steeper) the waterfall performance is. This is mainly because of the nature of iterative decoding and has little to do with the potential improvement in distance properties for longer interleavers.

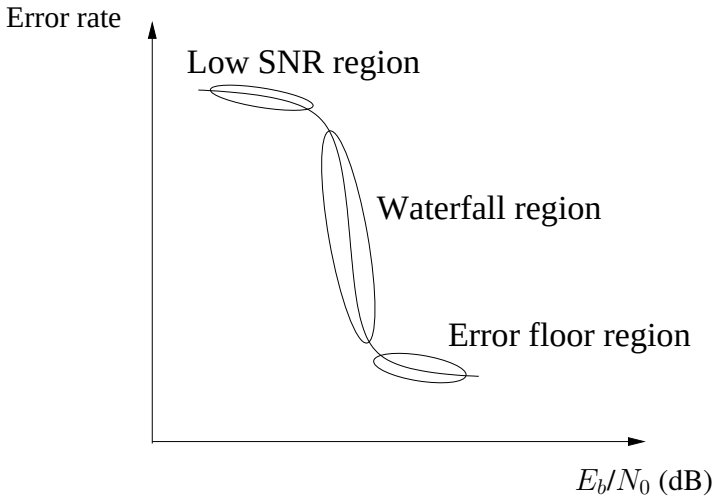


Figure 3.2: The three typical distinct regions of error correction performance of an iterative process.

- At high SNR, the error performance curve starts to flatten severely. Further improvements in the error performance require a significant increase in the SNR. The region associated with this severe flattening of the error performance curve is known as the error flare or error floor region. The better the distance properties, the lower the error flare is. In other words, high minimum distances with low multiplicities are important for lowering the error flare.

In this thesis work, convergence speed analysis will be conducted particularly for the waterfall region.

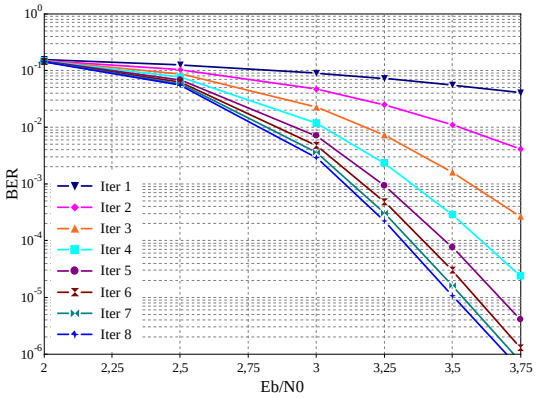
3.3.1 TBICM-SSD and TBICM-ID-SSD Error Correction Performance

Before starting the presentation of our studies on convergence speed analysis, this sub-section gives some reference BER curves in order to compare and appreciate the error correction performance with and without feedback loop to the demapper. Fig. 3.3 presents the results of different BER simulations for TBICM-SSD and TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. QPSK and QAM64 modulation schemes are selected with $R_c = \frac{1}{2}$ and R_c=$\frac{2}{3}$ respectively. These results show clearly how the TBICM-ID-SSD receiver mode outperforms the TBICM-SSD mode in terms of BER performance (i.e. to reach 10^{-5} BER, TBICM-SSD requires an $Eb/N0$ with 0.25 dB more).

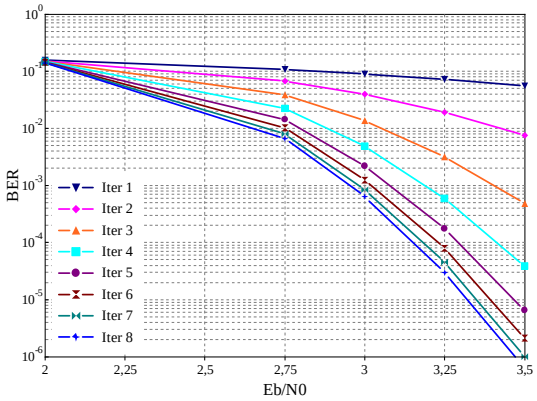
3.3.2 EXIT Chart Block Diagram

EXIT charts [54] are used as a useful tool for a clear and thorough analysis of the convergence speed. They were first proposed for parallel concatenated codes, and then extended to other iterative processes.

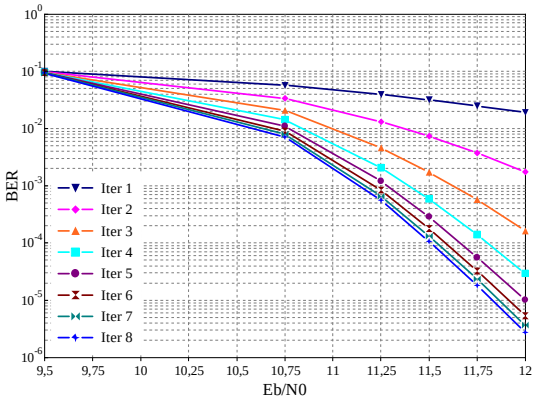
Fig. 3.4a shows the EXIT chart block diagram for TBICM-ID-SSD. For this iterative demapping receiver with turbo decoding (two iterative processes), the response of the two SISO decoders is plotted while taking into consideration the SISO demapper with updated inputs and outputs. In this scheme, IA_1, IA_2, IE_1, IE_2 are used to designate the *a priori* and extrinsic mutual information respectively for $\texttt{DEC}_1$ and $\texttt{DEC}_2$. In fact, mutual information designates the quantity that measures the



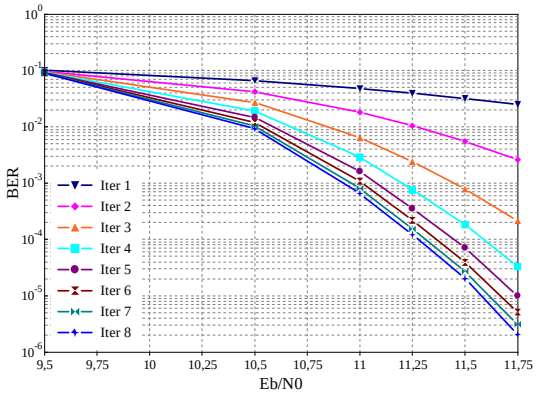
(a) TBICM-SSD: QPSK with R_c=$\frac{1}{2}$



(b) TBICM-ID-SSD: QPSK with R_c=$\frac{1}{2}$



(c) TBICM-SSD: QAM64 with R_c=$\frac{2}{3}$



(d) TBICM-ID-SSD: QAM64 with R_c=$\frac{2}{3}$

Figure 3.3: BER performance simulations for TBICM-SSD and TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. Different system configurations (QPSK and QAM64 modulation schemes with R_c=$\frac{1}{2}$ and R_c=$\frac{2}{3}$) are considered.

mutual dependence of the source bit and its correspondent LLR value. Iterations start without *a priori* information ($IA_1 = 0$ and $IA_2 = 0$). Then, mutual extrinsic information IE_1 of $\mathtt{DEC}_1$ is fed to $\mathtt{DEC}_2$ as mutual *a priori* information IA_2 and vice versa, i.e. $IE_1 = IA_2$ and $IE_2 = IA_1$. Since this EXIT chart analysis is asymptotic, long information frame size should be assumed.

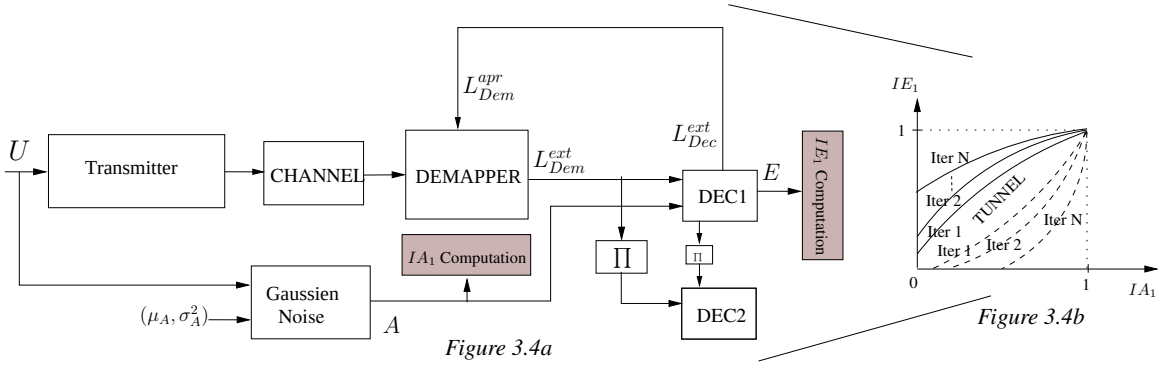


Figure 3.4: EXIT chart block diagram for turbo demodulation with turbo decoding.

The transfer function of the turbo decoder is represented by the two-dimensional chart as follows (Fig. 3.4b). One SISO decoder component is plotted with its input on the horizontal axis and its

output on the vertical axis. The other SISO component is plotted with its input on the vertical axis and its output on the horizontal axis. The iterative decoding corresponds to the trajectory found by stepping between the different curves. For a successful decoding, there must be a clear path (tunnel) between the curves so that iterative decoding can proceed from 0 to 1 mutual extrinsic information.

The *a priori* information available at the demapper input improves the BER at its output. The resulting iterative demapping scheme is equivalent to a demapper without *a priori* input at a higher value of $Eb/N0$. Having a changing value of $Eb/N0$ at the input of the decoder every demapper iteration, the computation of the mutual extrinsic information IE for the turbo decoder should, as a result, also be performed per demapping iteration (Fig. 3.4b). Hence, for better convergence, the tunnel should enlarge with every demapping iteration. The SISO decoder is represented by its transfer function:

$$IE = T(IA, E_b/N_0) \tag{3.11}$$

In fact, authors in [55] have suggested that the *a priori* input A=$\{a_1, a_2, \ldots, a_N\}$ (Fig. 3.4a) to the constituent decoder $\texttt{DEC}_1$ can be modeled by applying an independent Gaussian random variable z_A with variance σ_A^2 and mean zero in conjunction with the known transmitted information bits U=$\{u_1, u_2, \ldots, u_N\}$. N designates the number of source bits per frame. Decoder extrinsic information is denoted by E=$\{e_1, e_2, \ldots, e_N\}$. *A priori* information a_n can be expressed as follows.

$$a_n = \mu_A u_n + z_A \text{ where } \mu_A = \frac{\sigma_A^2}{2} \tag{3.12}$$

For each IA value, σ_A^2 can be computed using the following equation [56].

$$\sigma_A \approx (-\frac{1}{H_1} log_2(1 - IA^{\frac{1}{H_3}}))^{\frac{1}{2H_2}} \tag{3.13}$$

where H_1=0.3073, H_2=0.8935, and H_3=1.1064. Finally, the extrinsic mutual information IE is computed [57].

$$IE = 1 - \frac{1}{N} \sum_{n=1}^{N} log_2(1 + \exp^{-u_n e_n}) \tag{3.14}$$

3.3.3 Effects of Constellation Rotation

Extensive analysis of the effects of the constellation rotation for different system parameters (modulation orders, code rates, erasure probabilities) has been conducted. Fig. 3.5 and Fig. 3.6 illustrate 2 EXIT chart simulations for two different system configuration with and without erasure. QAM64 with R_c=$\frac{4}{5}$ and QPSK with R_c=$\frac{6}{7}$ are considered. E_b/N_0 values are chosen from the E_b/N_0 interval located in the waterfall region.

The plain curves correspond to the EXIT charts for the case with rotated constellation, while the dashed curves correspond to the case with no rotation. Furthermore, the red curves correspond to non iterative demapping. Applying demapping iterations corresponds to the other colored curves in the EXIT charts of Fig. 3.5 and 3.6.

In these two figures, we observe that the EXIT tunnel is wider for the rotated case than the one without. Furthermore, the tunnel is limited to that of one demapping iteration for the latter case. Thus, making more demapping iterations will not affect the convergence speed of non rotated constellation configurations. However the tunnel is enlarging until three demapping iterations using the rotated constellation.

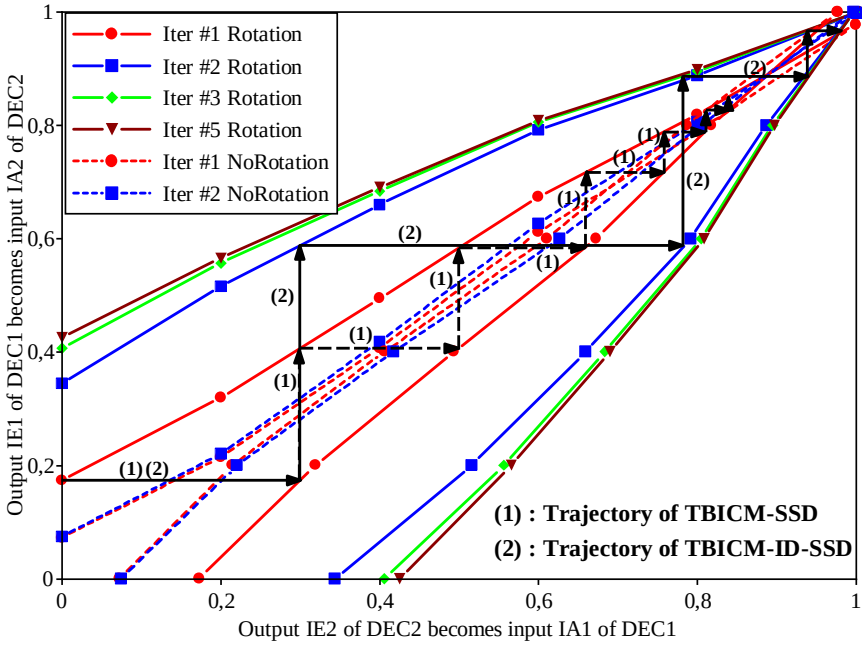


Figure 3.5: EXIT chart analysis at E_b/N_0= 22 dB of the double-binary turbo decoder for iterations to the QAM64 demapper. $R_c = \frac{4}{5}$ is considered for transmission over Rayleigh fast-fading channel with erasure probability P_ρ=0.15.

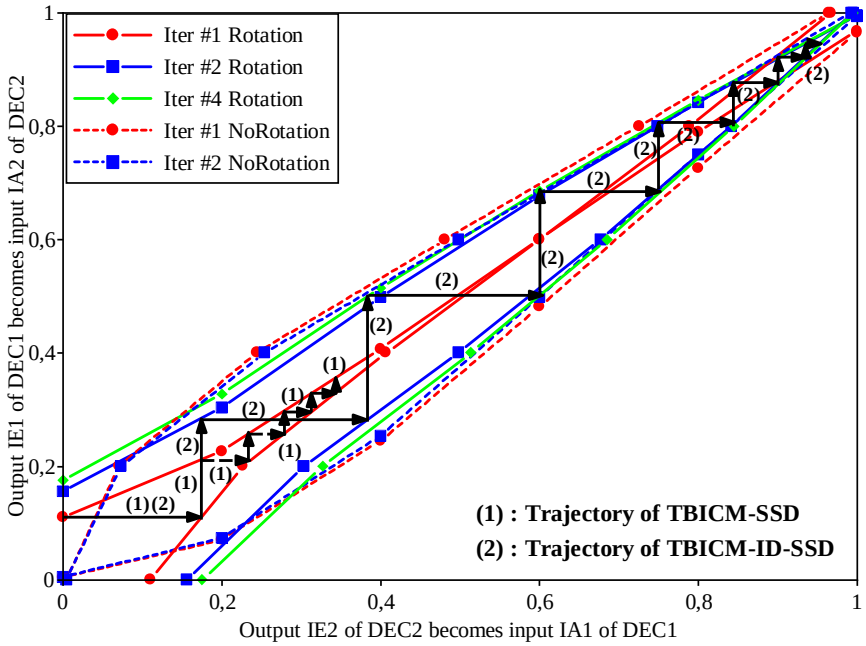


Figure 3.6: EXIT chart analysis at E_b/N_0= 7 dB of the double-binary turbo decoder for iterations to the QPSK demapper. $R_c = \frac{6}{7}$ is considered for transmission over Rayleigh fast-fading channel without erasure.

In Fig. 3.5, we can see that TBICM-SSD EXIT charts show a need of more than 6 turbo decoding iterations to attain convergence following the trajectory (1). Whereas 4 demapping iterations are sufficient following the trajectory (2). Moreover, Fig. 3.6 shows that the tunnel is blocked (convergence can not be attained) for the TBICM-SSD case. Whereas 8 demapping iterations are required to achieve the convergence for the TBICM-ID-SSD case.

Thus, in case of TBICM-ID-SSD, the iteration scheduling which optimize the convergence is

the one that enlarge (improve) the EXIT tunnel as soon as possible. Analyzing the different tunnel curves in the EXIT figure shows that the tunnel is enlarging for each demapping iteration. Thus, the optimized scheduling is to execute only one turbo decoding iteration for each demapping iteration and then step forward to the next demapping iteration (enlarge the EXIT tunnel). This scheduling is the one adopted implicitly in [19]. Note that after the third demapping iteration, only a slight improvement in convergence is observed. Similar results have been found for all considered modulation orders, code rates and erasure coefficients. This result will be used in section 3.4 in order to reduce the number of demapping iterations for TBICM-ID-SSD.

3.3.4 Effects of Bits-to-Symbol Allocation Scheme

In order to analyze the impact of the bits-to-symbol allocation scheme on the convergence speed of TBICM-SSD and TBICM-ID-SSD, two BICM S-random [58] interleavers are considered. The first one S_1 protects parity bits as well as systematic bits without any particular priority. The second one S_2 provides more error protection to systematic bits than parity. Fig. 3.7 and Fig. 3.8 illustrate 2 EXIT chart simulations for QAM16 modulation scheme, R_c=$\frac{1}{2}$ and Rayleigh fast-fading channel with and without erasures respectively. E_b/N_0 values are chosen from the E_b/N_0 interval located in the waterfall region.

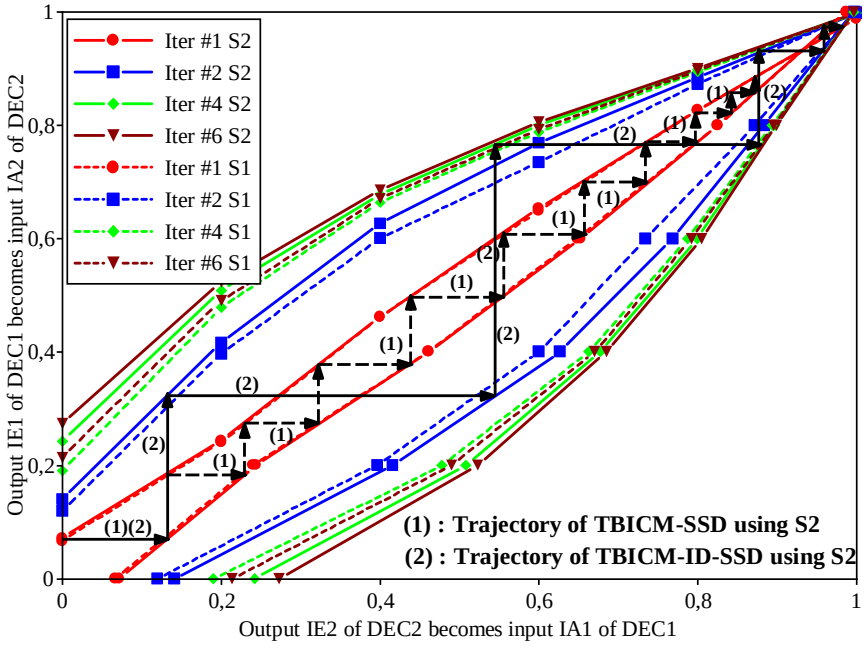


Figure 3.7: EXIT chart analysis at E_b/N_0= 14 dB of the double-binary turbo decoder for iterations to the QAM16 demapper. $R_c = \frac{1}{2}$ and Rayleigh fast-fading channel with erasure probability P_ρ=0.15 are considered.

The plain curves correspond to the EXIT charts for the case with interleaver S_2, while the dashed curves correspond to the case with S_1. Furthermore, the red curves correspond to non iterative demapping. Applying demapping iterations corresponds to the other colored curves in the EXIT charts.

We can see from Fig. 3.7 and Fig. 3.8 that the S_2 allocation scheme associated with turbo code outperforms, in terms of convergence speed, the first one S_1 in the waterfall region: The tunnel is wider whe S_2 is used. Thus the convergence speed is accelerated.

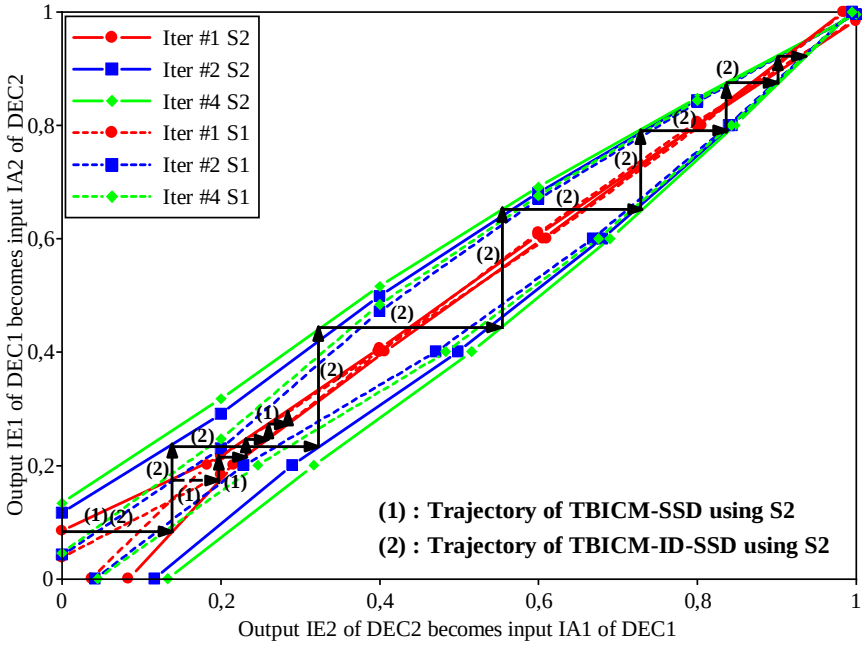


Figure 3.8: EXIT chart analysis at E_b/N_0= 5.5 dB of the double-binary turbo decoder for iterations to the QAM16 demapper. $R_c = \frac{1}{2}$ and Rayleigh fast-fading channel without erasure are considered.

3.3.5 Effects of Max-Log-MAP Algorithm

Fig. 3.9 illustrates the impact of using the Max-Log-MAP algorithm rather than the original MAP algorithm on the convergence speed of the turbo demodulation with turbo decoding receiver for the QAM16 modulation scheme, $R_c = \frac{1}{2}$, E_b/N_0= 5.5 dB, and P_ρ=0.

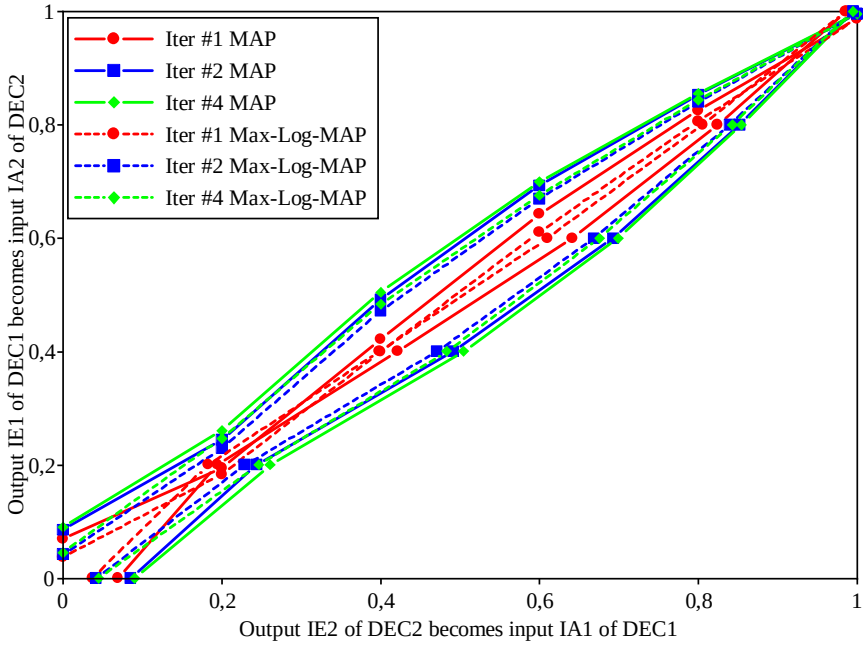


Figure 3.9: EXIT chart analysis at E_b/N_0= 5.5 dB of the double-binary turbo decoder for iterations to the QAM16 demapper. $R_c = \frac{1}{2}$ and Rayleigh fast-fading channel without erasure are considered.

The corresponding Max-Log-MAP tunnel , as shown in Fig. 3.9, is slightly reduced in comparison

to the MAP tunnel. However, for the rest of this thesis the Max-Log-MAP approximation will be considered as it offers a considerable complexity reduction solution for the MAP algorithm.

3.4 Reducing the Number of Demapping Iterations in TBICM-ID-SSD

As mentioned in the previous section, the typical optimized profile of iterations is the one that applies one demapping iteration for each turbo decoding iteration. In this profile, reducing the number of turbo demapping iterations will reduce the total number of iterations for the turbo decoder.

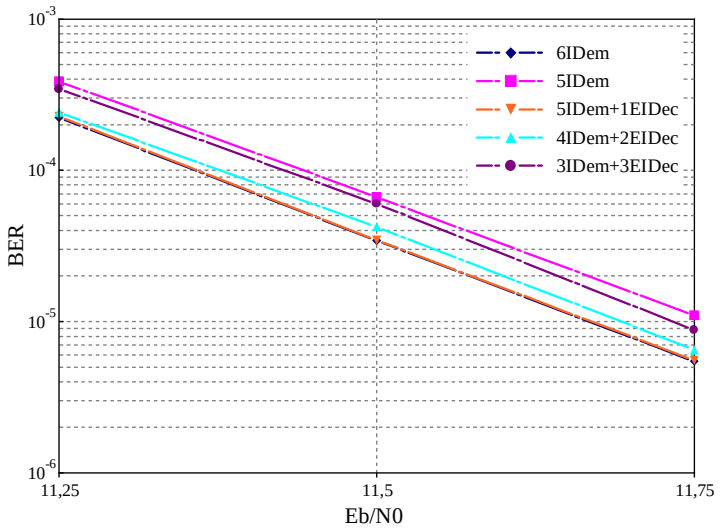


Figure 3.10: BER performance comparison for TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. QAM64 modulation scheme and $R_c = \frac{2}{3}$ are considered.

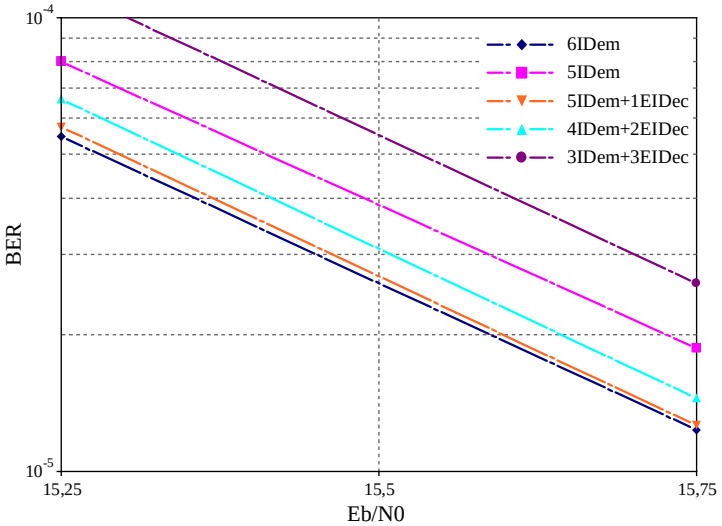


Figure 3.11: BER performance comparison for TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel with erasure probability P_ρ=0.15. QAM16 modulation scheme and $R_c = \frac{4}{5}$ are considered.

3.4.1 Proposed TBICM-ID-SSD Scheduling

Various constructed EXIT charts with different parameters show that after a specific number of demapping iterations, only a slight improvement is predicted. As an example, in Fig. 3.5 decoder transfer functions coincide with each other after 3 demapping iterations. However, one can notice that turbo decoding iterations must continue until that the two constituent decoders agree with each other. Thus, the number of demapping iterations can be reduced without affecting error rates, while keeping the same total number of turbo decoding iterations. This constitutes the basis for our proposed iteration scheduling.

In fact, to keep the same number of iterations for the decoder unaltered, one turbo decoding iteration is added after the last iteration to the demapper for each eliminated demapping iteration. Fig. 3.10 (QAM64, $R_c = \frac{2}{3}$, P_ρ=0) and Fig. 3.11 (QAM16, $R_c = \frac{4}{5}$, P_ρ=0.15) simulate six demapping iterations performing one turbo decoding iteration for each. Hence, six turbo decoding iterations are performed in total. This scheme is denoted as $6IDem$.

With the proposed iteration scheduling, $5IDem$+$1EIDec$ designates five demapping iterations (one turbo decoding iteration is applied for each) followed by one extra turbo decoding iteration.

Referring to Fig. 3.10 and Fig. 3.11, error rates associated to $6IDem$ and $5IDem$+$1EIDec$ show almost same performances, while one feedback to the demapper is eliminated in the latter scheme. Similarly, for $4IDem$+$2EIDec$, two feedbacks to the demapper are eliminated. A slight loss of 0.025 dB is induced. Eliminating more demapping iterations will cause significant performance degradation. Error rates performance of $3IDec$+$3EIDec$ is closer to $5IDem$ than to $6IDem$.

Modulation scheme	Performance loss (dB)	
	Without Erasure $R_c = 6/7 \text{ –▷ } R_c = 1/2$	With Erasure $R_c = 6/7 \text{ –▷ } R_c = 1/2$
QPSK	0.02 –▷ 0.03	0.02 –▷ 0.05
QAM16	0.04 –▷ 0.06	0.04 –▷ 0.08
QAM64	0.05 –▷ 0.08	0.07 –▷ 0.12
QAM256	0.07 –▷ 0.10	0.09 –▷ 0.15

Table 3.1: Performance loss for different modulation schemes and code rates after 2 omitted demapping iterations over Rayleigh fast-fading channel with and without erasure.

In fact, $4IDem$+$2EIDec$ represents the most optimized curve for the $6IDem$ performance scheme. EXIT charts do not agree with this consideration at the first sight, three demapping iterations followed by five extra turbo decoding iteration were sufficient to do the same correction as eight demapping iterations. EXIT charts are based on average calculations as many frames are simulated. The three demapping iterations represents the average number of demapping iterations needed to be sure that the two constituent decoders agree with each other. Making more demapping iterations will provide more error correction. Further simulations show performance loss of 0.02 dB to 0.1 dB and 0.02 dB to 0.15 dB for no erasure and erasure events respectively when the proposed scheduling is applied. Table 3.1 summarizes the reduced performance loss for different code rates and constellation orders after omitting two demapping iterations. These values were investigated in the waterfall region for the worst case (minimum required number of three demapping iterations) corresponding to $3IDem$+$2EIDec$ in comparison to $5IDem$. Note that for the error floor region, simulations (Fig. 3.12 as an example, which corresponds to Fig. 3.10 with larger Eb/N_0 margin) show almost identical BER performance if applying more than 3 demapping iterations. Furthermore, it is worth noting that with a limited-diversity channel model, omitting 2 demapping iterations leads to even lower performance loss than those of Table 3.1 for a fast-fading channel model. In fact, one

demapping iteration with high-diversity channel model leads to more error correction compared to one iteration executed with limited-diversity one. Thus, omitting demapping iterations for the former channel has higher impact on error rate performance. Conducted simulations with block-fading channel model have confirmed this result.

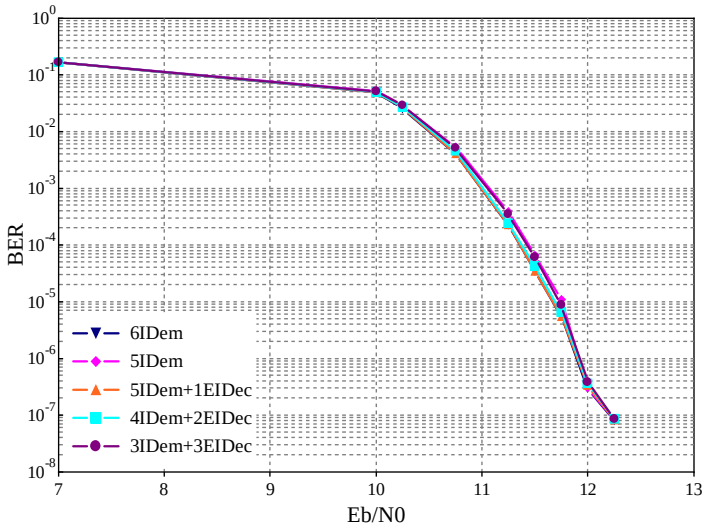


Figure 3.12: BER performance comparison for TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. QAM64 modulation scheme and $R_c = \frac{2}{3}$ are considered.

Using this technique, latency and complexity issues caused by the TBICM-ID-SSD are reduced. Two feedbacks to the demapper with the associated delays, computations, and memory accesses are eliminated. It is worth noting that the proposed new scheduling does not have any impact on the receiver area (logic or memory). This scheduling is applied on a TBICM-ID-SSD receiver and proposes a complexity reduction in "temporal dimension" (which impacts power consumption, throughput, and latency). Complexity reductions will be evaluated and discussed in the next section.

3.4.2 SISO Demapping and SISO Decoding Complexity Evaluation

The main motivation behind the conducted convergence speed analysis and the proposed technique for reducing the number of iterations is to improve the receiver implementation quality. In order to appreciate the achieved improvements, an accurate evaluation of the complexity in terms of number and type of operations and memory access is required. Such complexity evaluation is fair and generalized as it is independent from the architecture mode (serial or parallel) and remains valid for both of them. In fact, all architecture alternatives should execute the same number of operations (serially or concurrently) to process a received frame. In this section, we consider the two main blocks of the TBICM-ID-SSD system configuration which are the SISO demapper and the SISO decoder. The proposed evaluation considers the low complexity algorithms presented in subsection 3.2.2 and subsection 2.2.2.

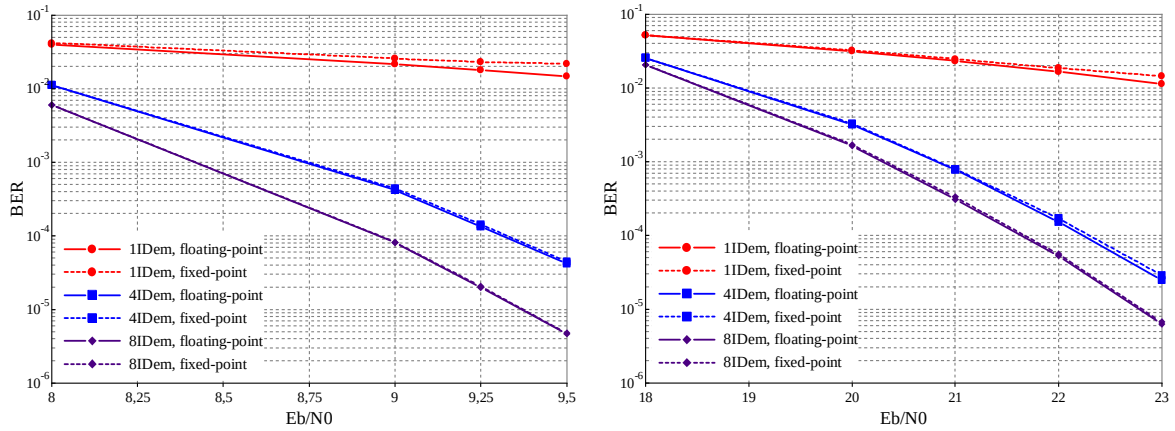
3.4.2.1 SISO Demapping and SISO Decoding Typical Quantization Values

A typical fixed-point representation of channel inputs and various metrics is considered. Table 3.2 summarizes the typical total number of required quantization bits for each parameter of the SISO demapper and SISO decoder [48].

	Parameter	Number of bits
SISO demapper	Received complex input $(x^I_{r,q}, x^Q_{r,q})$	(10,10)
	Coeff. Fading / Variance (h'_q/σ)	8
	Constellation complex symbol $(s^I_{r,j}, s^Q_{r,j})$	(8,8)
	Euclidean distance A_q	19
SISO decoder	Received 4 LLRs	4×5
	Branch metric γ_k	10
	State metric α_k, β_k	10
	Extrinsic information L^{ext}_{Dec}	10

Table 3.2: SISO demapping and SISO decoding typical quantization values.

Using this quantization, Fig. 3.13 plots two sets of floating-point vs fixed-point BER performance curves for TBICM-ID-SSD. Two modulation schemes, QPSK and QAM64, and different number of iterations are considered with R_c=$\frac{4}{5}$ and P_ρ=0.15. As we can see from this figure, considering the quantization of Table 3.2 provides almost the same BER as for the floating-point reference performance.



(a) QPSK **(b)** QAM64

Figure 3.13: Floating-point vs Fixed-point BER performance comparison for TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fast-fading channel with erasure probability P_ρ=0.15. QPSK and QAM64 modulation schemes are considered respectively for 1, 4, and 8 TBICM-ID-SSD iterations. R_c=$\frac{4}{5}$.

3.4.2.2 Complexity Evaluation of the SISO Demapper

The complexity of SISO demapping depends on the modulation order (section 3.2). We will now consider the equations of subsection 3.2.2 to compute: (1) the required number and type of arithmetic computations and (2) the required number of read memory accesses ($load$) and write memory accesses ($store$). The result of this evaluation is summarized in Table 3.3 and explained below. We use the following notation $operation$(NbOfBitsOfOperand1,NbOfBitsOfOperand2) for arithmetic operations, and $load$(NbOfBits)/$store$(NbOfBits) for read/write memory operations. Thus, $add(8, 10)$ indicates an addition operation of two operands; one quantized on 8 bits and the second on 10 bits. Similarly, $load(8)$ indicates a read access memory of 8-bit word length.

1) Euclidean distance computation

The Euclidean distance computation unit (equation (3.7)) can be represented as in Fig. 3.14. Quanti-

SISO rotated demapper with *a priori* input	Computation units	Number and Type of operations per modulated symbol per turbo demapping iteration
	Euclidean distance	$2^M Add(18,18) + 2^{M+1} Sub(8,10) + 2^{M+1} Mul(18,18) + 2^{M+1} Mul(8,10) + 2load(10) + (1+2^{M+1})load(8)$
	a priori adder	$(2^M-2)\{E[\frac{M-1}{2}]Add(8,8) + E[\frac{M-1}{4}]Add(9,9) + E[\frac{M-1}{8}]Add(10,10) + MSub(8,11) + MSub(11,19)\} + Mload(8) + (2^M-2)load(M)$ For QPSK $M(2^M-2)Sub(11,19) + Mload(8) + (2^M-2)load(M)$
	Minimum finder	$MSub(8,8) + M2^M Sub(19,19) + Mstore(8)$
SISO double-binary turbo decoder	Computation units	Number and Type of operations per coded symbol per turbo decoding iteration
	Branch metric	$4Add(5,5) + 38Add(5,10) + 4Sub(5,5) + 8load(5) + 6load(10)$
	State metric	$64Add(10,10) + 48Sub(9,9) + 8store(10)$
	extrinsic information	$32Add(10,10) + 32Sub(9,9) + 9Sub(10,10) + 3Mul(4,10) + 8load(10) + 5store(10)$

Table 3.3: SISO demapping and SISO decoding complexity computation summary.

zation values are taken from [48].

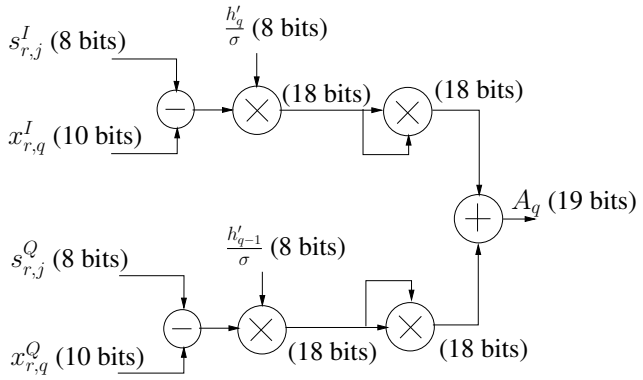


Figure 3.14: Euclidean distance computation unit

The computation of the Euclidean distance A_q implies the following operations for each modulated symbol (input of the demapper):

- One $load(8)$ to access the fading channel coefficient normalized by the channel variance ($\frac{h'_q}{\sigma}$).
- Two $load(10)$ to access the channel symbols $x^I_{r,q}$ and $x^Q_{r,q}$.
- For each one of the 2^M symbols of the constellation ($s^I_{r,j}$, $s^Q_{r,j}$):
  - Two $load(8)$ to access the constellation symbols $s^I_{r,j}$ and $s^Q_{r,j}$.
  - Two $Sub(8,10)$ to compute ($x^I_{r,q} - s^I_{r,j}$) and ($x^Q_{r,q} - s^Q_{r,j}$).
  - Two $Mul(8,10)$ to multiply the result with the channel coefficients $\frac{h'_q}{\sigma}$ and $\frac{h'_{q-1}}{\sigma}$.
  - Two $Mul(18,18)$ to compute the square of the above results.
  - One $Add(18,18)$ to realize the sum of the two Euclidean distance terms (I and Q).

2) A priori adder

The *a priori* adder computation unit (equation (3.9)) can be represented as in Fig. 3.15.

The computation of the *a priori* information $B_{p,q}$ implies the following operations for each modulated symbol (input of the demapper):

- M $load(8)$ to access the *a priori* informations $L^{apr}_{Dem}(c_{p,q})$.
- For each one of the 2^M symbols of the constellation ($s^I_{r,j}$, $s^Q_{r,j}$), except two symbols corresponding to all zeros and all ones:

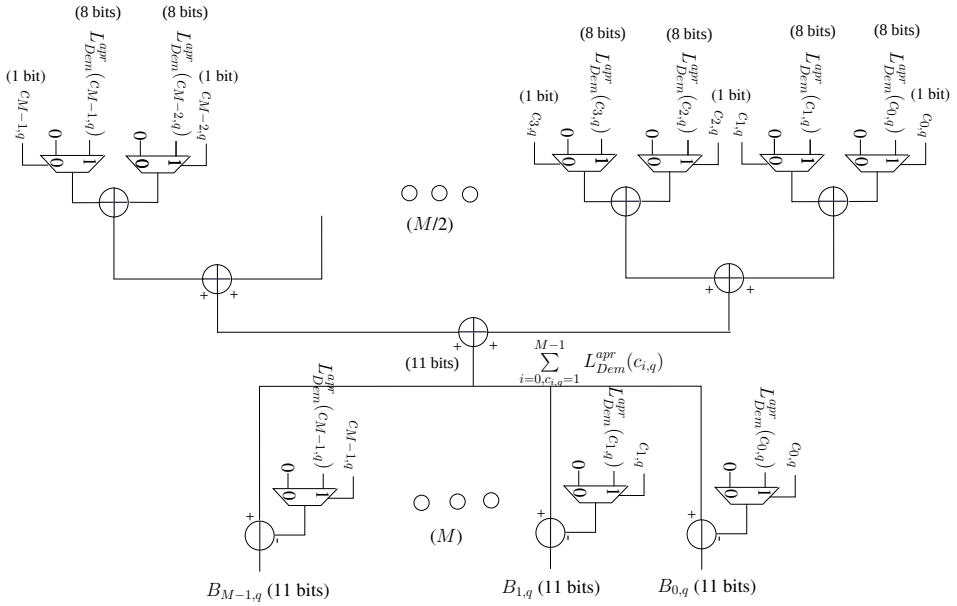


Figure 3.15: A priori adder computation unit

- One $load(M)$ to access constellation symbol bits $c_{i,q}$. $i = 0, 1, \ldots, M-1$ (equation (3.9)).
- One addition of M *a priori* information to compute $\sum_{i=0,c_{i,q}=1}^{M-1} L_{Dem}^{apr}(c_{i,q})$ of equation (3.9). $L_{Dem}^{apr}(c_{i,q})$ are quantized on 8 bits as shown in Table 3.2. This addition of M operands is equivalent to the sum of the following 2-input addition operations:
  - $E[\frac{M-1}{2}]$ $Add(8,8)$ to realize the sum of the couples of $L_{Dem}^{apr}(c_{i,q})$. Results are quantized on 9 bits.
  - $E[\frac{M-1}{4}]$ $Add(9,9)$ to realize the sum of the couples of the results above. Results are quantized on 10 bits.
  - $E[\frac{M-1}{8}]$ $Add(10,10)$ to realize the final 2-input addition of the results above. Note that $E[\frac{M-1}{8}]$ is equal to 0 except for QAM64 and QAM256 where it is equal to 1. The result is quantized on 11 bits.
    $E[x]$ represents here the ordinary rounding of the positive number x to the nearest integer.
- M $Sub(8,11)$ to subtract the LLR of the specific p^{th} bit and thus obtain $B_{p,q}$.
- M $Sub(11,19)$ to realize $A_q - B_{p,q}$ (equation (3.8)).

However, for the simple QPSK modulation the above operations can be simplified as only 2 LLRs exist for one modulated symbol. In fact, in equation (3.9) there is no need to execute an addition followed by a subtraction of the same LLR. Thus, the total number of required arithmetic operations in this case is $M(2^M - 2)$ $Sub(11,19)$ = 4 $Sub(11,19)$.

3) Minimum finder

The minimum finder computation unit can be represented as in Fig. 3.16.

The computation of the two minimum finders of equation (3.8) implies the following operations for each one of the M bits per modulated symbol:

- 2^M $Sub(19,19)$ to realize the two min operations of equation (3.8).

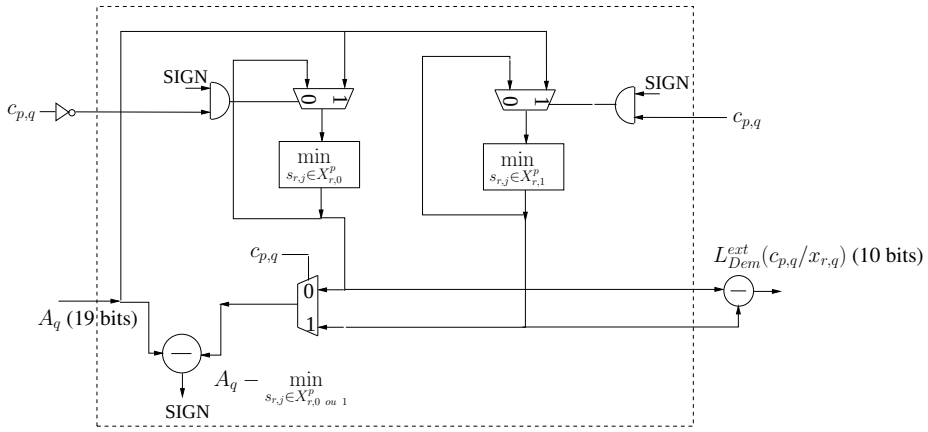


Figure 3.16: Minimum Finder for One LLR

- One $Sub(8,8)$ to subtract the above found 2 minimum values resulting in the demapper extrinsic information.
- One $store(8)$ to store the extrinsic information value.

3.4.2.3 Complexity Evaluation of the SISO Decoder

The SISO decoder complexity corresponds to the following 3 principal computations: branch metric, state metric, and extrinsic information computations. As for the SISO demapper, the result of the complexity evaluation is summarized in Table 3.3 and explained below. As stated before, the considered turbo code is an 8-state double-binary one. At the turbo decoder side, each double-binary symbol should be decoded to take a decision over the 4 possible values $(00, 01, 10, 11)$.

1) Branch metrics (γ)

The computation of the branch metrics of equation (2.11) implies the following operations for each coded symbol (input of the decoder):

- 4 $load(5)$ to access systematic and parity LLRs.
- 3 $load(10)$ to access demapper normalized extrinsic informations.
- 2 $Add(5,5)$ and 2 $Sub(5,5)$ to compute systematic and parity branch metrics γ_{11}^{Sys}, γ_{10}^{Sys}, γ_{11}^{Parity} and γ_{10}^{Parity}.
- 19 $Add(5,10)$ to compute branch metrics γ_k and $\gamma_k^{Sys} + \gamma_k^{Parity}$.

The operations above should be multiplied by 2 to generate forward and backward branch metrics.

2) State metrics (α,β)

The computation of the state metrics of equation (2.9) and equation (2.10) implies the following operations for each coded symbol (input of the decoder):

- 32 $Add(10,10)$ to compute $\alpha_{k-1}(s^{'}) + \gamma_k(s^{'}, s)$ for the 32 trellis transitions (8-state double-binary trellis).

- 24 $Sub(9,9)$ to realize the 8 max (4-input) operations of equation (2.9). In fact, 1 max (N-input) can be implemented as N-1 max (2-input) operations, and 1 max (2-input) is equivalent to 1 Sub (2-input).
- 8 $store(10)$ to store the computed state metrics (only for left butterfly).

The operations above should be multiplied by 2 to generate forward α and backward β state metrics.

3) Extrinsic information (z)

The computation of the extrinsic information of equation (2.13) implies the following operations for each coded symbol (input of the decoder):

- 8 $load(10)$ to access state metric values.
- 32 $Add(10,10)$ to compute the second required addition operation in equation (2.12) for the 32 trellis transitions.
- 28 $Sub(9,9)$ to realize the 4 max (8-input) operations of equation (2.12).
- 4 $Sub(10,10)$ to subtract symbol-level intrinsic information from the computed soft value (generating symbol-level extrinsic information).
- 8 $Sub(9,9)$ and 4 $Sub(10,10)$ to realize the 8 max (2-input) operations and compute 4 bit-level (systematic and parity) extrinsic information as demapper *a priori* information (equations (2.14) and (2.15)). This computation is done only for one of the two SISO decoders.
- 4 $store(10)$ to store the computed bit-level (systematic and parity) extrinsic information.
- 3 $Sub(10,10)$ to normalize symbol-level extrinsic information by subtracting the one related to decision 00.
- 3 $Mul(4,10)$ to multiply the symbol-level extrinsic information by a scaling factor SF (equation (2.13)).
- 3 $store(10)$ to store the computed $\texttt{DEC}_1$ symbol-level extrinsic information as $\texttt{DEC}_2$ *a priori* information.

3.4.2.4 Complexity Normalization

The above conducted complexity analysis exhibits different arithmetic and memory operation types and operand sizes. In order to provide a fair evaluation of the improvement in complexity with the technique proposed in section 3.4.1, complexity normalization is necessary.

Arithmetic operations ($n_2 \geq n_1$)	Normalized arithmetic operations
1 $Add(n_1,n_2)$	$0.5 \times (n_1+n_2-1)Add(1,1)$
1 $Sub(n_1,n_2)$	$0.5 \times (n_1+n_2)Add(1,1)$
1 $Mul(n_1,n_2)$	$[(n_1-1)(n_2-1)+1-0.5 \times n_1]Add(1,1)$

Table 3.4: Arithmetic operations normalization in terms of $Add(1,1)$.

For arithmetic operations, normalization can be done in terms of 2-input one bit full adders ($Add(1,1)$). Each one of the additions, subtractions, and multiplications can be converted into an equivalent number of $Add(1,1)$. For additions and subtractions, half (HA) and full adders (FA equivalent to $Add(1,1)$) of 2 operands of one bit for each are used and generalized for operand sizes

SISO rotated demapper with *a priori* input	Computation units	Number and Type of operations per modulated symbol per turbo demapping iteration
	Euclidean distance	$358.75\times2^{M+1}Add(1,1)$ + $load(28+2^{M+4})$
	a priori adder	$(2^M-2)\{7.5E[\frac{M-1}{2}]$+ $8.5E[\frac{M-1}{4}]$ + $9.5E[\frac{M-1}{8}]$ + $24.5M\}Add(1,1)$ + $load(8M)$ + $load(M(2^M-2))$ For QPSK $15M(2^M-2)Add(1,1)$ + $load(8M+M(2^M-2))$
	Minimum finder	$(8+19\times2^M)MAdd(1,1)$ + $store(8M)$
SISO double-binary turbo decoder	Computation units	Number and Type of operations per coded symbol per turbo decoding iteration
	Branch metric	$304Add(1,1)$ + $load(100)$
	State metric	$1040Add(1,1)$ + $store(80)$
	extrinsic information	$760Add(1,1)$ + $load(80)$ + $store(50)$

Table 3.5: SISO demapping and SISO decoding complexity computation summary after normalization.

n_1 and n_2. Obtained formulas are summarized in Table 3.4 with simple, yet accurate, analysis of all corner cases.

Fig. 3.17 shows 2 examples of addition and multiplication operations normalization. Two operands are considered. The first operand is quantized on n_1=2 bits, while the second operand is quantized on n_2=4 bits. The first example (Fig. 3.17(a)) corresponds to the addition operation normalization. 1 $Add(2,4)$ operation can be normalized to 1 HA + 1 HA + 1 FA + 1 HA operations. However, 1 HA can be considered as 0.5 FA. Hence, 2.5 FA (2.5 $Add(1,1)$) are required. Similarly for the second example (Fig. 3.17(b)). 1 $Mul(2,4)$ operation can be normalized to 1 HA + 1 FA + 1 FA + 1 HA operations. Hence, 3 FA (3 $Add(1,1)$) are required. These results can be obtained directly from Table 3.4 by putting the values of n_1 and n_2 in the corresponding equation for each considered operation.

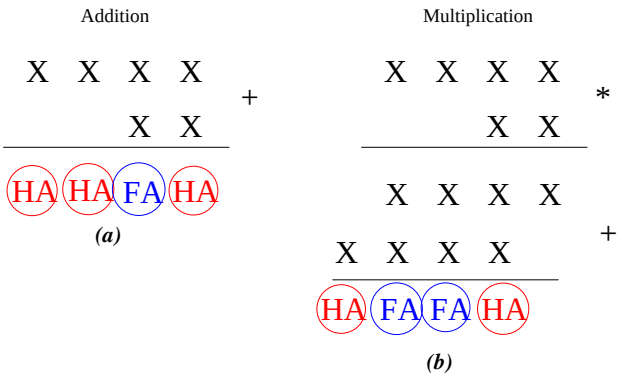


Figure 3.17: Complexity normalization examples: (a) Addition operation (b) Multiplication operation.

Similarly, multiplication operations are normalized using successive addition operations. Memory access operation of m word of size n are normalized to one memory access operation of $m \times n$ bits.

Applying the proposed complexity normalization approach to Table 3.3 leads to the results summarized in Table 3.5.

3.4.3 Discussions and Achieved Improvements

This section evaluates and discusses the achieved complexity reductions using the proposed iteration scheduling of TBICM-ID-SSD at different modulation orders and code rates. As concluded in section 3.4.1, two demapping iterations can be eliminated while keeping the number of turbo decoding iterations unaltered. Overall, this will lead to a reduction corresponding to two times the execution of the SISO demapping function. Besides the fact that the obtained results will depend on the modulation order and code rate, a third parameter should be considered regarding the iterative demapping implementation choice. In this regard, two configurations should be analyzed. In the first configuration,

denoted CASE 1, the Euclidean distances are re-calculated at each demapping iteration. While in the second configuration, denoted CASE 2, the computation of the Euclidean distances are done only once, at the first iteration, then stored and reused in later demapping iterations. Thus, CASE 1 implies higher arithmetic computations, however less memory access, than CASE 2.

Using the normalized complexity evaluation of Table 3.5, achieved improvements comparing $4IDem\_2EIDec$ to $6IDem$ for all configurations are summarized in Table 3.6. In the following we will explain first how these values are computed and then discuss the obtained results.

3.4.3.1 Complexity Reduction Ratio G_1

The complexity reduction ratio (G_1) is defined as the ratio of the difference in complexity between the original (C_{IDem}) and the new proposed ($C_{NEW-IDem}$) TBICM-ID-SSD scheduling to the complexity of TBICM-ID-SSD. It corresponds to the complexity reduction ratio of using the proposed scheduling rather than the original one. G_1 can be expressed as follows:

$$G_1 = \frac{C_{IDem} - C_{NEW-IDem}}{C_{IDem}} \tag{3.15}$$

If the original TBICM-ID-SSD configuration requires y iterations to process a frame composed of N_{MSymb} modulated symbols (equivalent to N_{CSymb} coded symbols), the original TBICM-ID-SSD complexity C_{IDem} can be computed by the following expression.

$$C_{IDem} = C^{-}_{dem}(M).N_{MSymb} + (y-1).C^{+}_{dem}(M).N_{MSymb} + y.C_{dec}.N_{CSymb} \tag{3.16}$$

where $C^{-}_{dem}(M)$ designates the complexity of processing one modulated symbol, which depends on the constellation size, without taking into consideration the *a priori* computation (first iteration). $C^{+}_{dem}(M)$ designates the complexity of processing one modulated symbol taking into consideration the *a priori* computation. C_{dec} designates the complexity of processing one coded symbol.

In fact G_1 corresponds to the ratio between the complexity of two SISO demapping executions and the complexity of the original TBICM-ID-SSD configuration. Hence, G_1 can be computed as:

$$G_1 = \frac{2.C^{+}_{dem}(M).N_{MSymb}}{C^{-}_{dem}(M).N_{MSymb} + (y-1).C^{+}_{dem}(M).N_{MSymb} + y.C_{dec}.N_{CSymb}} \tag{3.17}$$

Considering the code rate R_c and the number of bits per symbol M, the relation between the number of double-binary coded symbols (N_{CSymb}) and the corresponding number of modulated symbols (N_{MSymb}) can be written as follows.

$$\begin{aligned} N_{MSymb} &= \frac{\nabla.N_{CSymb}}{M.R_c} \\ &= \alpha.N_{CSymb} \text{ where } \alpha = \frac{\nabla}{MR_c} \end{aligned} \tag{3.18}$$

∇ is the number of bits per information symbol. ∇=2 for the double-binary case.

α_{max}=2 for ∇=2, M=2 and R_c=$\frac{1}{2}$. α_{min}=$\frac{7}{24}$≈0.292 for ∇=2, M=8 and R_c=$\frac{6}{7}$.

Converting in equation (3.17) the number of modulated symbols into equivalent coded symbols using equation (3.18), we obtain the following equation.

$$G_1 = \frac{2.C^{+}_{dem}(M)}{C^{-}_{dem}(M) + (y-1).C^{+}_{dem}(M) + \frac{1}{\alpha}.y.C_{dec}} \tag{3.19}$$

3.4.3.2 Achieved Improvements

This last equation has been used to obtain individually the complexity reductions in terms of arithmetic, read memory access, and write memory access operations of Table 3.6, for y=6. For CASE 1, results show increased benefits in terms of number of arithmetic operations (up to 32.4%) and read memory accesses (up to 29.9%) with higher modulation orders. This can be easily predicted from equation (3.19) as the value of C^{+}_{dem} increases with the constellation size. The equation shows also that the higher the code rate is, lower the benefits are.

On the other hand, the improvement in write memory access (3.7% for $R_c = 1/2$ and 2.2% for $R_c = 6/7$) is low and constant for all modulation orders. In fact, in Table 3.6 the single memory $store$ term which depends on the modulation order is $store(8.M)$ for the minimum finder computation. This term is required per modulated symbol and when converted to the equivalent number per coded symbol (equation (3.18)) for a fixed code rate a constant value, independent from M, is obtained.

Similar behavior is shown for CASE 2, except for two points. The first one concerns the improvements in arithmetic operations and read memory accesses. In fact, compared to CASE 1, this configuration implies less arithmetic and more memory access operations which lead to less benefits for the former and more benefits for the latter (equation (3.19)). The second point concerns the improvement in write memory access. In fact, besides the term $M \times 8bits$, a value of 19×2^M is required only for the first iteration to store the 2^M Euclidean distances quantized on 19 bits each. This added value is much higher in comparison to the reduced $M \times 8bits$ write memory access. Therefore the improvement in write access memory operations will be less for higher constellation sizes (down to 1.2%).

It is worth noting that applying the proposed scheduling combined with an early stopping criteria might diminish the benefit from the scheduling, but at the cost of an additional complexity.

Modulation scheme	CASE1 (With recomputed Euclidean distances)						CASE2 (With stored Euclidean distances)					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	Complexity Reduction			Complexity Reduction			Complexity Reduction			Complexity Reduction		
	$arith$	$load$	$store$	$arith$	$load$	$store$	$arith$	$load$	$store$	$arith$	$load$	$store$
QPSK	19.9%	12.8%	3.7%	15.4%	8.9%	2.2%	2.7%	11.5%	3.4%	1.8%	7.9%	2.1%
QAM16	25.8%	16.9%	3.7%	22.2%	12.5%	2.2%	10.8%	17.5%	3.1%	8.1%	13%	2%
QAM64	30.4%	24.4%	3.7%	28.6%	20.5%	2.2%	19.2%	25.4%	2.5%	16.9%	21.5%	1.7%
QAM256	32.4%	29.9%	3.7%	31.7%	27.8%	2.2%	24.2%	30.7%	1.5%	23.2%	28.7%	1.2%

Table 3.6: Reduction in number of operations, read/write access memory comparing "$4IDem\_2EIDec$" to "$6IDem$" for different modulation schemes and code rates.

3.5 Other Schedulings for TBICM-ID-SSD

In section 3.3, we have analyzed the convergence speed of the combined turbo demodulation with turbo decoding processes. EXIT charts have shown the optimized scheduling for applying one turbo code iteration for each demapping iteration.

However, a different scheduling for iterations can be applied. At each demapping iteration, only the decoding of a single component code of the turbo decoder can be performed. Hence, two demapping iterations are executed for a complete turbo decoding iteration. The a priori advantage of this system lies in the complexity reduction, since its complexity by iteration is given by the sum of the complexity of the demapper with only a single component decoder.

3.5.1 Scheduling Strategy

The idea of this scheduling is to avoid the execution of the two SISO decoder components per demapper iteration. Only a single component decoder (convolutional decoder) of the turbo decoder is executed after each demapping iteration. The extrinsic information of the current convolutional decoder is used to compose the *a priori* values for the demapper and as *a priori* information of the other component decoder used in the next iteration. Although only a single convolutional decoder is used per iteration, both convolutional decoders are processing through the iterations as in a turbo decoder.

First, the received symbols from the channel $\mathcal{X}_r$ and the demapper *a priori* information L_{Dem}^{apr}, which is equal to zero at the first iteration, feed the demapper to compute the extrinsic information L_{Dem}^{ext}. L_{Dem}^{ext} is then de-interleaved and used as *a priori* information for the first convolutional decoder $\texttt{DEC}_1$. $\texttt{DEC}_1$ computes its extrinsic information $L_{DEC_1}^{ext}$ which will be interleaved and used as *a priori* information for the demapper at the second iteration.

At the second iteration, the demapper extrinsic information is used as channel information for the second convolutional decoder $\texttt{DEC}_2$, and the extrinsic information taken from the component decoder $\texttt{DEC}_1$ in the previous iteration is used as *a priori* information for $\texttt{DEC}_2$. $\texttt{DEC}_2$ computes its extrinsic information $L_{DEC_2}^{ext}$ which will be fed jointed with $L_{DEC_1}^{ext}$ as *a priori* information to the demapper at the next iteration, and so one for the other iterations.

3.5.2 BER Performance Analysis

Fig. 3.18 illustrates the BER performance comparison between the original (presented in section 3.3.3 and the new2 TBICM-ID-SSD scheduling for QAM16 modulation scheme, $R_c = \frac{1}{2}$ and E_b/N_0=6 dB as a function of the number of demapping iterations. The coded frame size is taken N_{CSymb}=768 double-binary symbols.

It is shown from Fig. 3.18 that the new2 TBICM-ID-SSD scheduling provides bad BER performance values in comparison to the original scheduling. For the first demapping iteration, the difference in the BER performance comes from the fact that the new2 scheduling is performing only one component decoding instead of two for the original case. Moreover, a BER performance divergence is appeared for the second demapping iteration for the new2 scheduling. This is due to the difference in the demapper *a priori* information values provided by the two decoder components when performing the first feedback to the demapper (*a priori* information values from $\texttt{DEC}_1$ are presented, and *a priori* information values from $\texttt{DEC}_2$ are equal to zero).

After three demapping iterations, the new2 TBICM-ID-SSD scheduling shows a slow BER performance convergence in function of number of iterations in comparison to the original scheduling. This is due to the demapping process which takes into consideration only one *a priori* component decoder information per iteration.

In order to resolve the divergence problem for the new2 TBICM-ID-SSD scheduling and to improve the iterative receiver efficiency, we propose to modify this scheduling by combining it with the original one. The resulted scheduling will: (1) process with the original scheduling for a certain number of demapping iterations and then (2) it will process with the new2 scheduling for the

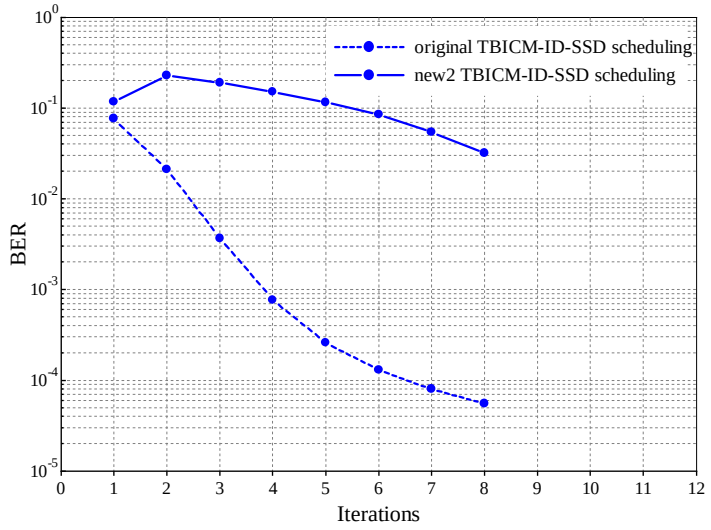


Figure 3.18: BER performance comparison between the original and the new2 TBICM-ID-SSD scheduling as a function of number of iterations for the transmission of 1536 information bits frame over Rayleigh fading channel without erasure. QAM16 modulation scheme, $R_c = \frac{1}{2}$ and E_b/N_0=6 dB are considered.

rest of the iterations. Taking the example of y demapping iterations performing with the original TBICM-ID-SSD scheduling, the modified-new2 scheduling will perform x ($x \prec y$) original demapping iterations and then z iterations performing with the new2 scheduling. z should be superior to $y - x$ ($z = y - x + \delta, \delta \succ 0$) in order to recover the BER performance loss while executing only one decoder component for each of the z iterations instead of two decoder components for the original scheduling.

This modified-new2 scheduling aims to provide less executions (less complexity) of SISO components decoder at the expense of additional execution (more complexity) of SISO demapping. In fact x should be chosen close to y ($y - x \leq 3$), otherwise there will be need of a high number of additional iterations performing with the new2 scheduling and thus increasing the complexity of the modified-new2 scheduling in comparison to the original scheduling. The case for $x = y - 1$ should not be also considered as only the last iteration (1 demapper and 2 decoder components execution) in the original scheduling is omitted and replaced by at least two new2 scheduling iterations (2 demapper and two decoder components execution). Hence, the resulted complexity will be superior to the complexity of the original scheduling.

3.5.3 Complexity Analysis

The main motivation behind the conducted modified-new2 scheduling proposition is to reduce the complexity of the TBICM-ID-SSD receiver while achieving the same BER performance values. In order to appreciate the modified-new2 scheduling benefits, a complexity analysis is required.

Let $C_{NEW2-IDem}$ be the TBICM-ID-SSD receiver applying the modified-new2 scheduling. It can be expressed as follows.

$$\begin{aligned} C_{NEW2-IDem} = {} & C^{-}_{dem}(M).N_{MSymb} + (x-1)C^{+}_{dem}(M).N_{MSymb} + xC_{dec}.N_{CSymb} \\ & + (y-x+\delta)C^{+}_{dem}(M).N_{MSymb} + (y-x+\delta)\frac{C_{dec}}{2}.N_{CSymb} \end{aligned} \quad (3.20)$$

The complexity reduction ratio G_2 for using the modified-new2 scheduling instead of the original one can be written as:

$$G_2 = \frac{C_{IDem} - C_{NEW2-IDem}}{C_{IDem}} \tag{3.21}$$

Using this equation and replacing C_{IDem} and $C_{NEW2-IDem}$ by their expressions from equations (3.16) and (3.20) and converting the number of modulated symbols into equivalent coded symbols using equation (3.18) lead to the following equation:

$$G_2 = \frac{a - b}{d} \tag{3.22}$$

where

$$\begin{aligned} a &= \frac{(y - x - 2\delta)}{2\alpha} C_{dec} \\ b &= \delta C^{+}_{dem}(M) \\ d &= C^{-}_{dem}(M) + (y-1)C^{+}_{dem}(M) + \frac{y}{\alpha}.C_{dec} \end{aligned}$$

a designates the reduced complexity for using the modified-new2 scheduling with less SISO decoding executions. b designates the added complexity for using the modified-new2 scheduling with more demapping executions. d designates C_{IDem} divided by α.

Referring to equation (3.22), the modified-new2 scheduling will provide less complexity in comparison to the original scheduling when $a \succ b$ in terms of arithmetic operations and read/write memory access.

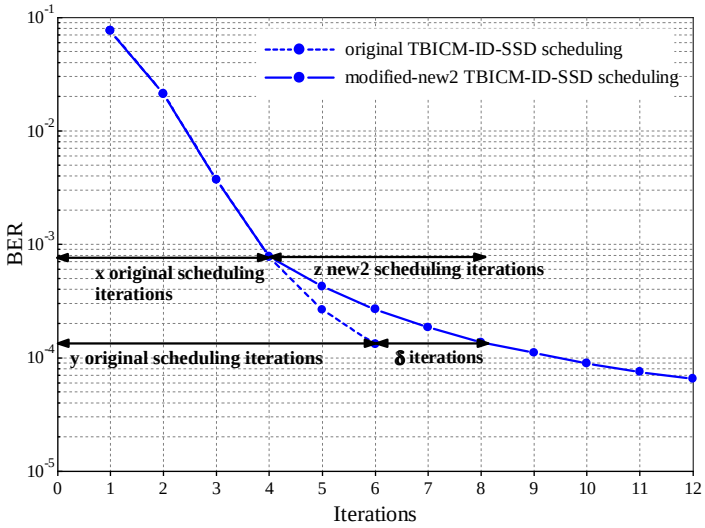


Figure 3.19: BER performance comparison between the original and the modified-new2 TBICM-ID-SSD scheduling as a function of number of iterations for the transmission of 1536 information bits frame over Rayleigh fading channel without erasure. QAM16 modulation scheme, R_c=$\frac{1}{2}$ and E_b/N_0=6 dB are considered.

As mentioned in the previous subsection, x should be chosen close to y. We choose for example x=$y - 2$. Fig. 3.19 illustrates the BER performance comparison between the original scheduling applying y=6 iterations and the modified-new2 scheduling applying x=4 original scheduling iterations

and several z new2 scheduling iterations. QAM16 modulation scheme, R_c=$\frac{1}{2}$ ($\alpha = 1$), E_b/N_0=6 dB and N_{CSymb}=768 double-binary symbols are considered.

Fig. 3.19 shows that x=4 and z=4 iterations are required for the modified-new2 scheduling to achieve a BER=$1,3.10^{-4}$ instead of 6 iterations for the original scheduling. This means that the modified-new2 scheduling is executing with the same total number of SISO decoder components (12 SISO decoder components executions), however two additional SISO demapping executions (8 SISO demapper executions instead of 6) are required. Thus, the modified-new2 scheduling requires higher complexity than the original one.

Various BER performance simulations comparison between these two scheduling have been plotted for different system configurations. Similar results are obtained, the modified-new2 scheduling is processing with higher complexity in comparison to the original scheduling. Hence, the modified-new2 scheduling will not be considered for the rest of this thesis.

3.6 Complexity Adaptive TBICM-ID-SSD Receiver

The objective of this section is to illustrate for which system configuration it is more interesting to use the TBICM-ID-SSD mode rather than TBICM-SSD. This means for which system configuration the complexity of TBICM-ID-SSD becomes lower than TBICM-SSD. To this end, the complexity of each mode in terms of the complexity of the SISO demapper and SISO decoder will be defined. The required number of iterations assuming identical complexity will be analyzed. Finally, a complexity analysis for identical BER performance will be presented.

It is worth to note that this section does not provide a comparison in terms of area, as the initial receiver is considered to perform both modes TBICM-SSD and TBICM-ID-SSD.

3.6.1 TBICM-SSD and TBICM-ID-SSD Complexity Expressions

If the TBICM-SSD mode requires x iterations to process a frame composed of N_{MSymb} modulated symbols, the complexity C_{IDec} for TBICM-SSD can be calculated as the sum of the complexity of one demapping process and x decoding processes.

$$C_{IDec} = C^{-}_{dem}(M).N_{MSymb} + xC_{dec}.N_{CSymb} \tag{3.23}$$

Regarding the complexity of TBICM-ID-SSD, we consider for the rest of this chapter the work of section 3.4.1 which proposes an original iteration scheduling by reducing two demapping iterations with reasonable performance loss of less than 0.15 dB for all configurations. The required number of iterations is denoted like in section 3.4.1 by $yIDem$+z$EIDec$, where z designates the extra decoding iterations. Thus, the complexity of the proposed scheduling $C_{NEW-IDem}$ can be computed as the sum of the complexity of y demapping processes and $(y + z)$ decoding processes.

$$C_{NEW-IDem} = C^{-}_{dem}(M).N_{MSymb} + (y-1)C^{+}_{dem}(M).N_{MSymb} + (y+z)C_{dec}.N_{CSymb} \tag{3.24}$$

3.6.2 Number of Iterations Analysis for Identical Complexity

The corresponding number of iterations if both modes TBICM-SSD and TBICM-ID-SSD have identical complexity can be analyzed. Identical complexity can be expressed as $C_{IDec} = C_{NEW-IDem}$.

Using this equality and replacing C_{IDec} and $C_{NEW-IDem}$ by their expressions from equations (3.23) and (3.24) lead to the following equation:

$$xC_{dec}.N_{CSymb} = (y-1)C^{+}_{dem}(M).N_{MSymb} + (y+z)C_{dec}.N_{CSymb} \tag{3.25}$$

This last equation allows to obtain the number of TBICM-ID-SSD iterations y=y_{Lim} corresponding to identical complexity for both modes. In fact, by replacing N_{MSymb} with equivalent number of N_{CSymb} (equation (3.18)) and by simplifying, equation (3.25) becomes:

$$y_{Lim} = \frac{(x-z)C_{dec} + \alpha.C^{+}_{dem}(M)}{C_{dec} + \alpha.C^{+}_{dem}(M)} \tag{3.26}$$

This equation can be used to compute individually y_{Lim} for identical arithmetic, identical read memory access or identical write memory access operations.

If we consider x=6, and for different modulation orders and code rates, Table 3.7 shows the required number of iterations y_{Lim} with no extra decoding iteration (z=0).

y_{Lim} can have positive values as well as negative values. Negative values mean that for the chosen configuration, TBICM-ID-SSD has always a higher complexity than TBICM-SSD. The positive values represent the limits for which performing less demapping iterations will lead to a lower complexity than TBICM-SSD, and the inverse is true. Hence, it might be possible to perform less y iterations ($y \prec y_{Lim}$) with less complexity while having the same error correction capability than TBICM-SSD. In fact, Table 3.7 shows that this last situation can potentially happen for QPSK and QAM16 configurations where y_{Lim} varies in a higher range (between 2.9 and 5.8) than QAM64 and QAM256 configurations (most y_{Lim} values are around 2 corresponding to identical arithmetic operations). This analysis will be extended in the next subsection taking into consideration error rate performance simulations.

Modulation scheme	CASE1 (With recomputed Euclidean distances)						CASE2 (With stored Euclidean distances)					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	y_{Lim}			y_{Lim}			y_{Lim}			y_{Lim}		
	arith	*load*	*store*	*arith*	*load*	*store*	*arith*	*load*	*store*	*arith*	*load*	*store*
QPSK	4.2	4.4	5.5	4.8	4.9	5.7	5.6	4.2	4.9	5.8	4.8	5.3
QAM16	2.9	3.9	5.5	3.6	4.6	5.7	4.1	3.4	4.4	4.7	4	5
QAM64	1.8	2.8	5.5	2.2	3.4	5.7	2.4	2.2	2.7	2.9	2.8	4
QAM256	1.3	1.7	5.5	1.4	2.1	5.7	1.4	1.5	−2.9	1.7	1.8	0.6

Table 3.7: Identical complexity (arithmetic operations, or read, or write access memory): Number of required demapping iterations for $x = 6$ and $z = 0$ for different modulation schemes and code rates.

3.6.3 Complexity Analysis for Identical Performance and Achieved Improvements G_3

It is known that for equal number of TBICM-SSD and TBICM-ID-SSD iterations, $xIDec$ and $yIDem$ respectively, iterative processing at the demapper side is shown to provide additional error correction [19]. Thus, for a considered number of x iterations, identical error rate performance results can be reached by using y iterations with $y < x$.

3.6.3.1 Complexity Analysis for a Chosen x

Fig. 3.20 shows a BER comparison between the two iterative modes TBICM-SSD and TBICM-ID-SSD for two configurations: (1) QPSK, R_c=$\frac{4}{5}$, P_ρ=0.15 and (2) QAM64, R_c=$\frac{2}{3}$, P_ρ=0. These

parameters are chosen to represent clearly the two sets of curves in the same figure, the same behavior is seen for other configurations. The BER for x=6 TBICM-SSD iterations and for different configurations, can be seen as the result of y=3 and y=4 TBICM-ID-SSD iterations for erasure and non erasure channel respectively. In fact, the TBICM-ID-SSD mode will provide better results for the erasure case than the one without. Hence, less demapping iterations are required for the former case to achieve the same performances as TBICM-SSD. However, using the results of section 3.4.1, the complexity of $4IDem$ could be reduced to $3IDem$+z$EIDec$ with $z = 1$. In fact, we have shown in section 3.4 that omitting only one demapping iteration while adding one extra turbo decoding iteration will keep the error rate performance almost identical for number of TBICM-ID-SSD iterations $y > 3$.

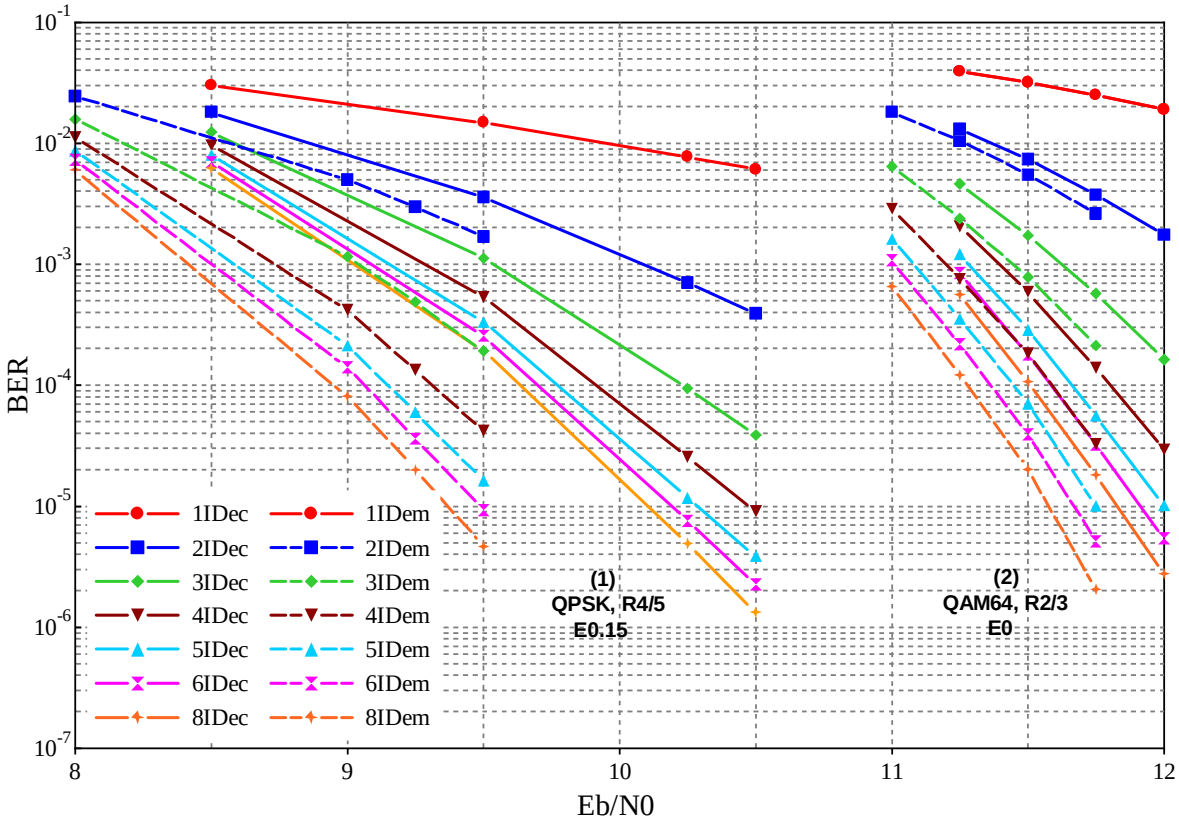


Figure 3.20: BER performance comparison between TBICM-SSD and TBICM-ID-SSD over iterations for the transmission of 1536 information bits frame over Rayleigh fading channel with and without erasure. Different modulation schemes are code rates are considered

On the other hand, Table 3.7 shows that for QPSK modulation, the minimum number of required TBICM-ID-SSD iterations y_{Lim} for all code rates and for identical required arithmetic operations as $6IDec$ is y_{Lim}=4.2 with $z = 0$. So using y=3<4.2 iterations will lead to less arithmetic complexity, meanwhile it has the same error correction capacity as illustrated in Fig. 3.20 for configuration (1).

Complexity improvements have been computed and summarized in Table 3.8 and Table 3.9. These tables resume the achieved improvements comparing $6IDec$ to $3IDem$+$1EIDec$ and $3IDem$+$0EIDec$ respectively for all configurations. In the following we will explain first how these values are computed and then discuss the obtained results.

The complexity reduction ratio (G_3) is defined as the ratio of the difference in complexity between the two iterative modes to the complexity of TBICM-SSD. It corresponds to the gain ratio of using TBICM-ID-SSD rather than TBICM-SSD. G_3 can be expressed as follows:

Modulation scheme	CASE1 (With recomputed Euclidean distances)						CASE2 (With stored Euclidean distances)					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	Complexity reduction			Complexity reduction			Complexity reduction			Complexity reduction		
	arith	*load*	*store*	*arith*	*load*	*store*	*arith*	*load*	*store*	*arith*	*load*	*store*
QPSK	13.5%	16.8%	28.6%	21.4%	23.5%	30.6%	28%	14%	19.1%	30.1%	21.8%	25%
QAM16	−15.6%	9.3%	28.6%	2.7%	18.9%	30.6%	10.7%	−3.5%	9.5%	19.2%	11.2%	19.3%
QAM64	−89.5%	−22.9%	28.6%	−50.9%	−1.5%	30.6%	−35.8%	−58.6%	−22.3%	−14%	−23.7%	0.6%
QAM256	−207.4%	−108.4%	28.6%	−158.5%	−62.3%	30.6%	−117.9%	−195.6%	−124.1%	−87.2%	−121.1%	−59.3%

Table 3.8: No Erasure channel: Reduction in number of operations, read/write access memory comparing "3$IDem$+1$EIDec$" to "6$IDec$" for different modulation schemes and code rates.

Modulation scheme	CASE1 (With recomputed Euclidean distances)						CASE2 (With stored Euclidean distances)					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	Complexity reduction			Complexity reduction			Complexity reduction			Complexity reduction		
	arith	*load*	*store*	*arith*	*load*	*store*	*arith*	*load*	*store*	*arith*	*load*	*store*
QPSK	28.9%	32.6%	45%	37.2%	39.6%	47%	43.3%	29.8%	35.4%	45.9%	38%	41.4%
QAM16	−1.6%	24.9%	45%	17.7%	34.9%	47%	24.7%	12.1%	25.9%	34.2%	27.2%	35.8%
QAM64	−78.8%	−8.6%	45%	−38.3%	13.7%	47%	−25.1%	−44.3%	−5.9%	−1.5%	−8.5%	17.1%
QAM256	−201.4%	−97.2%	45%	−150.4%	−49.3%	47%	−111.9%	−184.4%	−107.8%	−79%	−108.1%	−42.8%

Table 3.9: Erasure channel: Reduction in number of operations, read/write access memory comparing "3$IDem$+0$EIDec$" to "6$IDec$" for different modulation schemes and code rates.

$$G_3 = \frac{C_{IDec} - C_{NEW-IDem}}{C_{IDec}} \tag{3.27}$$

Using this equation and replacing C_{IDec} and $C_{NEW-IDem}$ by their expressions from equations (3.23) and (3.24) lead to the following equation:

$$G_3 = \frac{(x-y-z)C_{dec}.N_{CSymb} - (y-1)C^{+}_{dem}(M).N_{MSymb}}{xC_{dec}.N_{CSymb} + C^{-}_{dem}(M).N_{MSymb}} \tag{3.28}$$

By replacing N_{MSymb} with equivalent number of N_{CSymb} (equation (3.18)) and by simplifying, equation (3.28) becomes:

$$G_3 = \frac{(x-y-z)C_{dec} - \alpha(y-1)C^{+}_{dem}(M)}{xC_{dec} + \alpha.C^{-}_{dem}(M)} \tag{3.29}$$

This last equation has been used to obtain individually the complexity reduction ratios of Table 3.8 and Table 3.9 in terms of arithmetic, read memory access and write memory access operations. Positive values correspond to a decreasing in complexity when using the TBICM-ID-SSD, meanwhile negative values correspond to a an increasing in complexity.

In the following, we analyze the values of Table 3.8 which correspond to a no erasure channel. Similar behavior is seen in Table 3.9 for erasure channel.

For CASE 1, results show improvements in terms of number of arithmetic operations (up to 21.4%) and read access memory (up to 23.5%) for QPSK scheme. Higher modulation orders require the demapper to fetch symbols from higher constellation memory sizes, which lead to more complexity computations and memory accesses. An increasing in complexity is shown for QAM256 in terms of number of arithmetic operations (-207%) and read access memory (-108%). On the other hand, the improvements in write memory access (28.6% for $R_c = 1/2$ and 30.6% for $R_c = 6/7$) are positive for all modulations orders.

In fact, in the SISO demapper, write memory access is required only to store the extrinsic information which is composed of $M \times 8bits$. This term is required per modulated symbol and when

converted to the equivalent number per coded symbol (equation (3.18)) for a fixed code rate, a constant value independent from M is obtained.

Moreover, equation (3.29) shows that higher the code rate (lower α) is, higher the benefits are. This is due to the fact that the term multiplying α in the numerator is higher than the term multiplying α in the denominator. Table 3.8 confirms this idea.

Similar behavior is shown for CASE 2, except for two points. The first one concerns the improvements in arithmetic operations and read memory access. In fact, compared to CASE 1, this configuration implies less arithmetic and more memory access operations in the SISO demapper which lead to more benefits for the former operations and less benefits for the latter. The second point concerns the benefits in write memory access. In fact, besides the term $M \times 8bits$, a value of 19×2^M is required to store the 2^M Euclidean distances quantized on 19 bits each. Therefore the benefits in write access memory operations will be less for high constellation sizes.

Taking an example of QAM64 and code rate $\frac{6}{7}$ for CASE 1 with erasure, Table 3.9 shows an increasing in complexity in terms of arithmetic operations (-38.3%), meanwhile positive ratios are seen for read/write access memory. However, it should be noted that the number of required memory access are much less than the arithmetic operations. Thus, those latter are considered as the primary criteria for choosing between the two modes.

We can conclude from the results above that using TBICM-ID-SSD rather than TBICM-SSD for QPSK and QAM16 orders will lead to a significant complexity reduction for almost all code rates. The number of normalized arithmetic operations is reduced in a range between 28.9% and 45.9% for the QPSK configuration for using TBICM-ID-SSD rather than TBICM-SSD with 6 iterations over fading channel with erasures.

Finally, as the proposed adaptive iterative receiver targets to reduce the overall normalized processing complexity, this should lead a priori to improved power consumption, throughput and latency. However, analyzing the detailed improvements in terms of throughput and latency depends on the heterogeneous architecture and the parallelism degree of the considered demapper and decoder algorithms.

3.6.3.2 Complexity Analysis for Different Values of x

The second part of this study is to look to the reduction values for different values of x. To that end, and for presentation simplicity, we consider one system configuration which corresponds to QPSK, R_c=$\frac{4}{5}$ and erasure probability P_ρ=0.15.

Fig. 3.21 illustrates the BER performance for both modes as a function of the number of iterations at E_b/N_0=9.5 dB. From this figure, we obtain Table 3.10 which illustrates the equivalent y iterations for different x values for identical error rate performances.

x	1	2	3	4	5	6	7	8
y	1	1.7	2.2	2.5	2.75	2.9	2.95	3

Table 3.10: TBICM-SSD and TBICM-ID-SSD equivalent number of iterations for QPSK, R_c=$\frac{4}{5}$ and P_ρ=0.15.

Using Table 3.10 and equation (3.29), we obtain the complexity reduction curves of Fig. 3.22. Only CASE 2 is considered for presentation simplicity, however the results are similar for CASE 1. The curves of Fig. 3.22 show the variation of the benefits in number of arithmetic operations, read and write memory accesses as a function of the number of iterations x. In fact, Table 3.10 shows that for TBICM-SSD number of iterations x=1 the corresponding number of TBICM-ID-SSD iterations

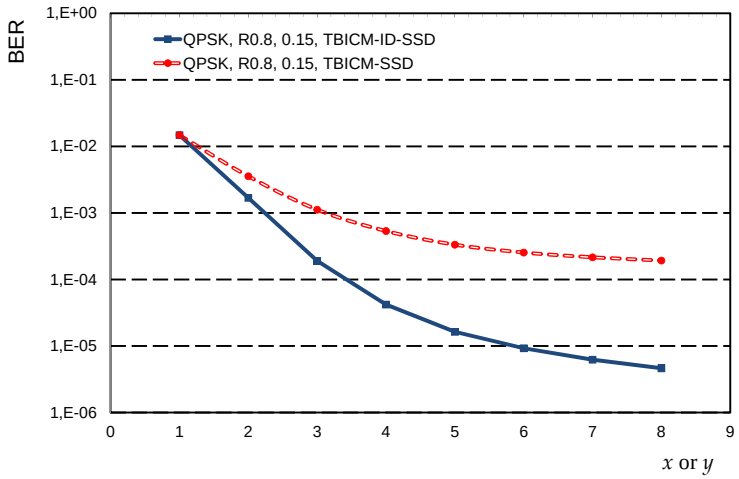


Figure 3.21: BER performance comparison between TBICM-SSD and TBICM-ID-SSD as a function of number of iterations for the transmission of 1536 information bits frame over Rayleigh fading channel with erasure probability equals to 0.15. QPSK modulation scheme with R_c=$\frac{4}{5}$ and E_b/N_0=9.5 dB are considered.

y=1. This corresponds to no feedback loop to the demapper, and thus, to identical complexity of the two modes TBICM-SSD and TBICM-ID-SSD. This result is illustrated by Fig. 3.22 where the complexity reduction ratio G_3=0 for x=1. For x=2, the complexity reduction in terms of arithmetic operations and read access memory is about 10% and 2% respectively. However, an increased need of write access memory is shown. This is due to the added complexity for storing the 2^M Euclidean distances computed at the first iteration. In fact, the difference in equivalent number of iterations x and y is not big enough to recover this memory write access overhead. However, for $x > 2$, this difference becomes significant and the complexity reduction ratio increases almost linearly with x to reach between 50% to 60% for x=8. This can be explained from Table 3.10 where increasing x will increase y but with less speed to attain identical error rate performances.

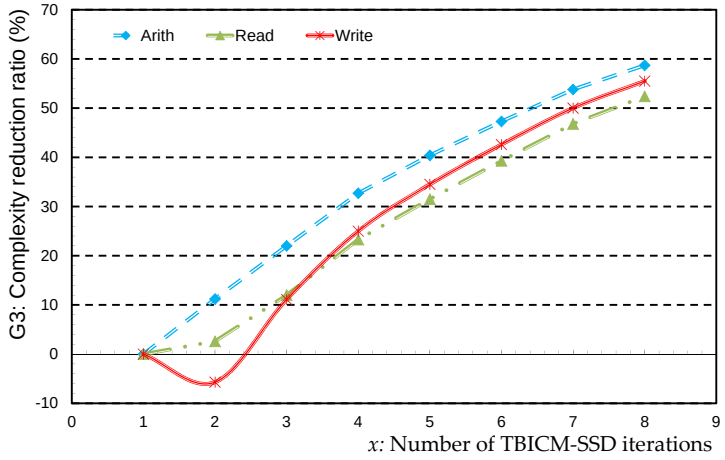


Figure 3.22: Complexity reduction over iterations for using TBICM-ID-SSD rather than TBICM-SSD for QPSK with R_c=$\frac{4}{5}$, P_ρ=0.15 and CASE 2.

3.7 Efficient Sizing of Heterogeneous Multiprocessor Receivers for TBICM-ID-SSD

Flexible baseband receivers gain the interest of many research efforts to enable the design of future multi-modes multi-standards terminals. A main challenge in this domain is to provide this flexibility with minimum overhead in terms of area, speed, and energy. In this regard, heterogeneous multiprocessor platforms are emerging as a promising implementation solution. However, the heterogeneity of such platforms makes it complex to find the required number of processors supporting a specific configuration (i.e. requirements level).

At the architecture level, many efforts are being conducted towards the design of flexible high throughput hardware platforms which can be configured to the required configuration. Homogeneous [59–61] and heterogeneous [62–66] approaches for flexible multi-standards platforms have been developed in the past years. The overall flexibility of the radio platform can be achieved through the flexibility of individual components at transmitter side (encoder, interleaver, mapper, etc.) and at receiver side (equalizer, demapper, de-interleaver, decoder, etc.). The high throughput requirement imposes often the efficient exploitation of different parallelism levels. In this context, multi-processor architecture [67–71] is a promising approach to reach high flexibility, high throughput and energy efficiency.

A configuration is defined by the communication parameters which are chosen in accordance with the application requirements and the environment in which the communication is established. Formal expressions which allow designers to optimize the receiver architecture by computing the required number of processors depending on each configuration can be proposed. This point is essential as it enables designers to formally explore potential architectures that will meet performance constraints. Such a solution significantly mitigates design exploration task which is a critical step in the design process and avoids the oversized platforms approach which is based only on worst use case.

This section investigates, in this context, the significant optimization potential both *at design-time* and *at run-time* regarding the selection of the most appropriate configuration. A formal representation of the architectural solution space which allows designers to find the minimum hardware configuration is proposed. The proposed approach is illustrated through a flexible multi-processor hardware platform for turbo demodulation with turbo decoding. This approach corresponds to a joint work with another Ph.D student, Vianney Lapotre.

3.7.1 Generic heterogeneous multiprocessor architecture model

Fig. 3.23 presents the generic architecture of a flexible multi-processor hardware platform for TBICM-ID-SSD. The aim of this platform is to provide a flexible and dynamic solution compared to existing ones (generally based on hardware accelerators) where designer can tune the number of resources both at design-time and at run-time. As it will be presented, such an approach allows the system meeting performance constraints without loosing its flexibility. These features will be mandatory for future communication systems. In Fig. 3.23, DemProc and DecProc perform demapping and decoding algorithms respectively. These two processors are characterized by their area, maximum frequency, and their performance defined by the number of cycles to demap or decode one modulated or coded symbol respectively. The platform integrates a communication interconnect that allows extrinsic information exchanges (between DecProcs themselves and between DecProcs and DemProcs).

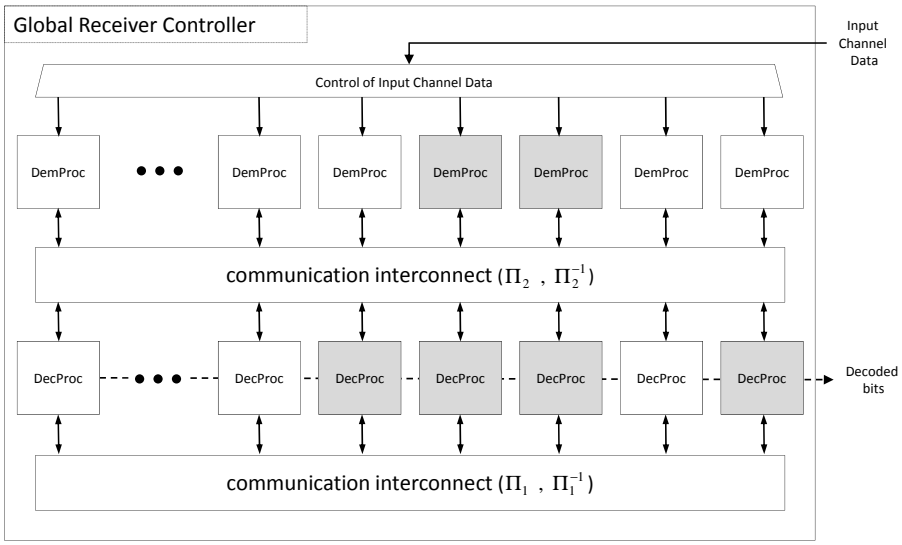


Figure 3.23: Generic architecture of the heterogeneous multiprocessor receiver. In this configuration example 2 DemProcs and 4 DecProcs are not used.

3.7.2 Formal representation of the architectural solution space

The generic architecture of Fig. 3.23 can be abstracted as two components: one demapper and one decoder. Each component uses several processors in parallel to perform the frame computation exploiting sub-bloc parallelism. These two components are serially connected. The time required to process one frame (T_{syst}) corresponds to the sum of the time required by the demapper (T_{dem}) and the time required by the decoder (T_{dec}) to execute all their iterations on the frame. It can be expressed as:

$$\begin{aligned} T_{syst} &= T_{dem} + T_{dec} \\ &= N_{MSymb}.T_{dem/symb} + N_{CSymb}.T_{dec/symb} \end{aligned} \tag{3.30}$$

where $T_{dem/symb}$ and $T_{dec/symb}$ represent respectively the time required by the demapper and the decoder to execute all their iterations on one modulated and coded symbol respectively. Hence, the system throughput ($D_{syst} = N_{CSymb}/T_{syst}$) can be expressed as:

$$D_{syst} = \frac{D_{dem}.D_{dec}.N_{CSym}}{N_{MSymb}.D_{dec} + N_{CSymb}.D_{dem}} \tag{3.31}$$

where $D_{dem} = 1/T_{dem/symb}$ and $D_{dec} = 1/T_{dec/symb}$ are the demapper and the decoder throughputs (in modulated and coded symbols respectively).

Converting the number of modulated symbols into equivalent coded symbols (equation (3.18)) in equation (3.31) gives the following system throughput expression.

$$D_{syst} = \frac{D_{dem}.D_{dec}}{D_{dem} + \alpha.D_{dec}} \tag{3.32}$$

D_{syst} is generally imposed by the application requirement. On the other hand, the throughput of the demapper and the decoder depend on the number of processors, the number of iterations, the number of clock cycles required to process on symbol, and the clock frequency. D_{dem} and D_{dec} can be expressed as follows.

$$D_{dem} = \frac{Nb_{demProc}.F_{dem}}{it_{dem}.cycles_{dem/symb}} \tag{3.33}$$

$$D_{dec} = \frac{Nb_{decProc}.F_{dec}}{2.it_{dec}.cycles_{dec/symb}} \tag{3.34}$$

where $Nb_{demProc}$ and $Nb_{decProc}$ designate respectively the number of demapping and decoding processors, it_{dem} and it_{dec} designate respectively the number of demapping and decoding iterations, $cycles_{dem/symb}$ and $cycles_{dec/symb}$ designate respectively the number of cycles required to demap and to decode one symbol, F_{dem} and F_{dec} designate respectively the demapper and the decoder clock frequency.

From equations (3.33) and (3.34), we can express $Nb_{demProc}$ and $Nb_{decProc}$ as follows.

$$Nb_{demProc} = K_{dem}.D_{dem} \tag{3.35}$$

$$Nb_{decProc} = K_{dec}.D_{dec} \tag{3.36}$$

where K_{dem} and K_{dec} depend on the system configuration and the processor parameters. They can be expressed as below.

$$K_{dem} = \frac{it_{dem}.cycles_{dem/symb}}{F_{dem}} \tag{3.37}$$

$$K_{dec} = \frac{2.it_{dec}.cycles_{dec/symb}}{F_{dec}} \tag{3.38}$$

It is worth noting that the linear increase in decoding throughput with the number of decoding processors is limited due to the sub-bloc initialization issue [72]. This limitation, which depends on the target frame size and code rate, should be considered in the platform sizing. However, this problem is not encountered in the demapping sub-bloc parallelism.

In order to establish a relation between the demapping time and the decoding time, we define the ratio n as follows.

$$n.T_{dem} = T_{dec} \tag{3.39}$$

From this equation we can obtain a relation between the demapper throughput D_{dem} and the decoder throughput D_{dec}.

n	$Nb_{demProc}$	$Nb_{decProc}$
0.25	40	44
0.75	56	21
1	64	18
1.25	72	16
1.75	88	14

Table 3.11: Architecture alternatives in function of n. Example for: 200 Mbps, QPSK, R_c=0.5, it_{dem}=it_{dec}=8, $cycles_{dem/symb}$=6, $cycles_{dec/symb}$=1.75 (except for the last iteraton which is equal to 0.75), F_{dec}=F_{dem}=300 MHz

$$\begin{aligned} n.\frac{N_{MSymb}}{D_{dem}} &= \frac{N_{CSymb}}{D_{dec}} \\ D_{dem} &= D_{dec}.n.\frac{N_{MSymb}}{N_{CSymb}} \\ D_{dem} &= D_{dec}.n.\alpha \end{aligned} \tag{3.40}$$

Putting equation (3.40) into equation (3.32), we deduce the expressions which connect the throughput of the system with the throughputs of the demapper and the decoder:

$$D_{dem} = \alpha.(n+1).D_{syst} \tag{3.41}$$

$$D_{dec} = \frac{n+1}{n}.D_{syst} \tag{3.42}$$

Replacing D_{dem} and D_{dec} in equations (3.35) and (3.36) by their expressions from equations (3.41) and (3.42) allows us to compute the number of demapper processors $Nb_{demProc}$ and the number of decoder processors $Nb_{decProc}$ required for a given configuration and a given n.

$$Nb_{demProc} = K_{dem}.\alpha.(n+1).D_{syst} \tag{3.43}$$

$$Nb_{decProc} = K_{dec}.\frac{n+1}{n}.D_{syst} \tag{3.44}$$

Table 3.11 illustrates how different values of n lead to different architecture alternatives, although all of them achieving the target throughput and supporting the target system configuration. $cycles_{dec/symb}$ is taken equal to 1.75 [73] (except for the last iteraton which is equal to 0.75). $cycles_{dem/symb}$ is taken equal to 4 [48] for the QPSK case. Depending on n we observe that the number of processors can vary from 44 to 14 for decoding and from 40 to 88 for demapping. It is essential, both at *design-time* and at *run-time*, to determine the value of n which optimizes the resources use. The optimization goal depends of designers priorities and could be for example the number of processors used for each possible configuration, the total area of the chip, the clock frequency for each type of processor, etc.

3.7.3 Area optimization

Heterogeneous processors have typically different areas and performances. One main optimization objective is to determine $Nb_{demProc}$ and $Nb_{decProc}$ in order to minimize the receiver area for a given

configuration. The total area of the receiver depends on n. It can be computed using the expression below.

$$A_n = A_{dem}.Nb_{demProc} + A_{dec}.N_{decProc} \tag{3.45}$$

where A_{dem} and A_{dec} represent respectively the area of one DemProc and one DecProc. Therefore, by putting equations (3.35), (3.36) and (3.40) into equation (3.45), A_n can be expressed as a function of K_{dem}, K_{dec} and D_{dec}.

$$A_n = (K_{dec}.A_{dec} + K_{dem}.A_{dem}.\alpha.n)D_{dec} \tag{3.46}$$

Finally, A_n can be expressed as a function of n by putting equation (3.42) into equation (3.46).

$$A_n = \frac{a.n^2 + b.n + c}{n} \tag{3.47}$$

where

$$\begin{aligned} a &= C_{dem}.A_{dem}.D_{syst}.\alpha \\ c &= C_{dec}.A_{dec}.D_{syst} \\ b &= a + c \end{aligned}$$

The derivative function of the equation (3.47) is then computed. Only one extremum (n_{ext}) is found. It can be computed as:

$$n_{ext} = \sqrt{\frac{c}{a}} = \sqrt{\frac{2.it_{dec}.cycles_{dec}.F_{dem}.A_{dec}}{it_{dem}.cycles_{dem}.F_{dec}.A_{dem}.\alpha}} \tag{3.48}$$

The second derivative function is also computed at n_{ext}. It shows a positive value corresponding to the minimum area (A_{next}) of the receiver. Finally, A_{next} can be computed as:

$$A_{next} = a + c + 2\sqrt{a.c} \tag{3.49}$$

Fig. 3.24 shows the variation of the area A_n in function of n, only one minimum A_{next} exists. Hence, for a given configuration, n_{ext} can be determined with equation (3.48). The number of DemProc and DecProc which minimizes the area are then computed using equations (3.43) and (3.44). These two equations give decimal numbers, hence the number of processors is rounded up to guarantee the throughput constraint.

This approach can be applied for TBICM-SSD (it_{dem}=1) and for TBICM-ID-SSD (it_{dec}=it_{dem}). Note that if the number of DecProc, computed in equation (3.44), is above the parallelism limit, this number should be saturated in accordance to the maximum level of available parallelism and the corresponding number of DemProc will be computed with respect to the throughput requirement.

Based on the set of equations above it is possible to analyze how the system can be tuned both at design-time and at run-time to meet performance requirements for a given configuration.

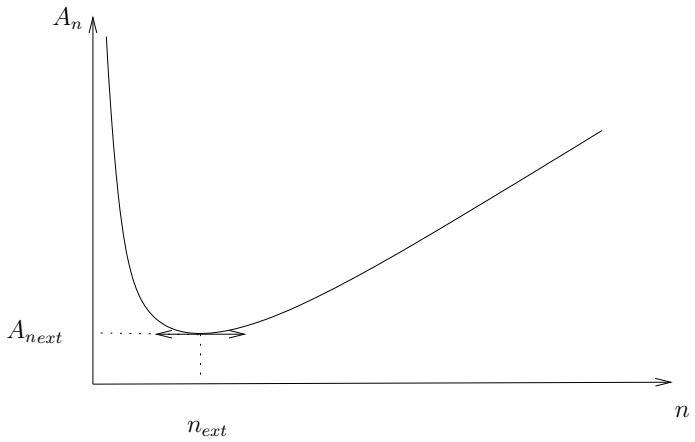


Figure 3.24: Flexible multi-processor hardware area with one extremum for iterative demodulation with turbo decoding.

3.8 Complexity Reduction of Shuffled Parallel TBICM-ID-SSD

In the previous sections of this chapter the TBICM-ID-SSD receiver was running in a serial mode. Demapper and decoder (in natural and interleaved domain) components are executed sequentially. Extrinsic values generated by the SISO demappers and SISO decoders are exchanged at the end of each component execution. Whereas in [49], the authors compared the serial mode to a full shuffle mode (shuffled turbo decoding with shuffled iterative demapping). In this latter mode, all SISO demappers and SISO decoders components are executed simultaneously exchanging extrinsic information as soon as created. This is a unique scheme introduced by the authors, in the context of high throughput receivers, to execute both the demapping and decoding tasks concurrently.

In fact, the SISO turbo decoding schemes presented in subsection 2.3 have illustrated three different schemes for applying the Max-Log-MAP decoding algorithm. The simplest scheme is the Forward-Backward where the state metric values are first computed, followed by extrinsic information computations. A parallel computation of both metrics in forward and backward direction, also known as the butterfly scheme (*B*), was also investigated. In this scheme, extrinsic values are only generated in the second half of the turbo decoding iteration. In order to improve the convergence of the shuffled turbo decoding, butterfly-replica (*B-R*) scheme is proposed. In this latter, the extrinsic values are generated continuously all along the iteration period.

This section analyzes the convenience of using an appropriate scheme to reduce the complexity of high parallel TBICM-ID-SSD receiver: full shuffled iterative receiver with multiple SISO decoders and SISO demappers. This analysis corresponds to a joint work with another Ph.D student, Oscar Sanchez.

3.8.1 Parallel Full Shuffled TBICM-ID-SSD Strategy

Fig. 3.25 shows the information exchanged between the multiple demapper and decoder components in parallel full shuffled iterative demapping with turbo decoding. I SISO demappers are used to process a frame of N_{MSymb} modulated symbols. Meanwhile, $2J$ SISO decoders are assigned to decode a frame of N_{CSymb} coded symbols. Intensive information exchange is carried out between the SISO decoders in natural and interleaved domain as well as among all SISO decoders and SISO demappers.

In this scheme, one shuffled demapping iteration can be defined as a simultaneous execution of one demapping process and one turbo decoding process.

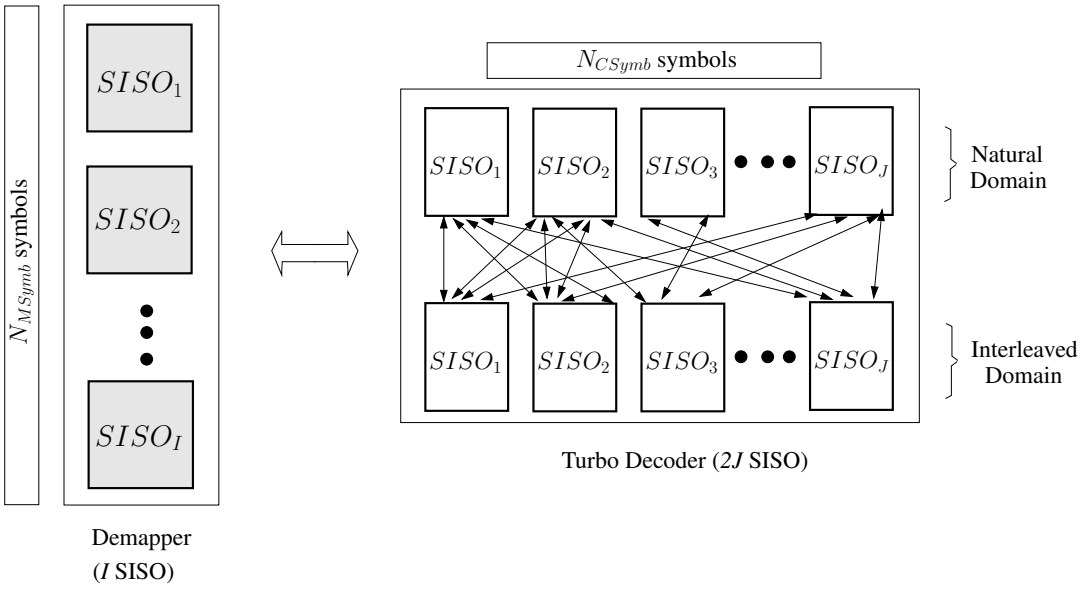


Figure 3.25: Parallel full schuffled iterative demapping with turbo decoding.

Let τ_{Dem} and τ_{Dec} be the throughput of the SISO demappers and SISO decoders respectively, *i.e.* the number of demapped symbols and decoded symbols processed per second. A parameter representing the symbol processing throughput ratio between the demapper and the decoder components is defined:

$$D_r = \tau_{Dec}/\tau_{Dem} \tag{3.50}$$

For the full shuffled receiver, the best configuration to reduce the latency is that where both demapping and decoding tasks finish at the same time [49]. In this efficient scheme, the demapper and decoder components will take the same time to process a whole frame, while exchanging information, during a complete iteration. In fact, the demapper and decoder tasks are different in nature. Moreover, different number of symbols are assigned for each component. Based on this idea, the required ratio between the number of demappers and decoders was given [49].

$$\frac{J}{I} = \frac{M \times R_c}{\nabla \times D_r} \tag{3.51}$$

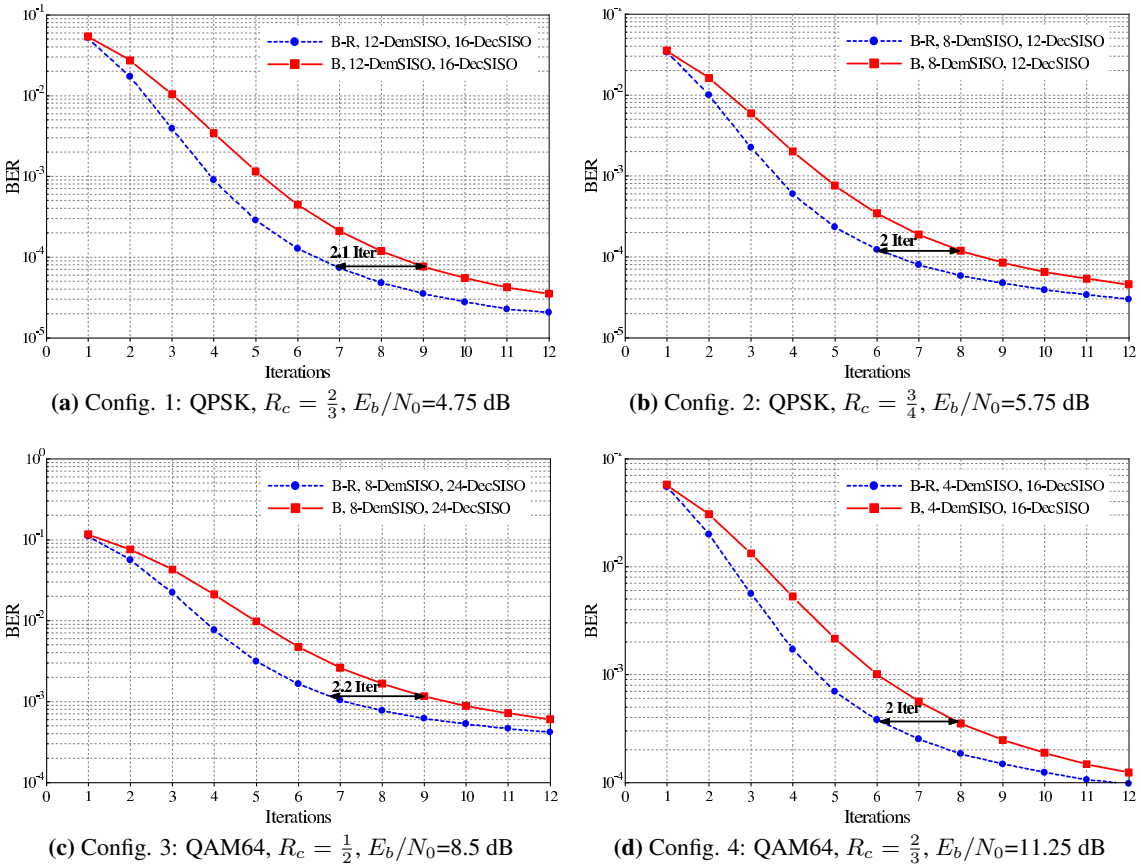
This last equation will be used for the rest of this section in order to configure the receiver with the required number of I SISO demappers and $2J$ SISO decoders.

3.8.2 Simulations and Achieved Improvements

This subsection analyzes the convenience of the *B* and *B-R* schemes for iterative demapping with turbo decoding receiver running in a full shuffle mode. For a fair comparison between the two schemes, bit error rate simulations and complexity evaluation in terms of arithmetic operations and memory accesses are presented.

3.8.2.1 Performance Simulations for Different Shuffled Turbo Decoding Schemes

A flexible software model for the whole shuffled system was developed. It supports different modulation schemes (QPSK, QAM16, QAM64 and QAM256), code rates (from $\frac{1}{3}$ to $\frac{6}{7}$) and a variable number of SISO demappers and SISO decoders. The Rayleigh fast-fading channel without erasure is considered.



(a) Config. 1: QPSK, $R_c = \frac{2}{3}$, E_b/N_0=4.75 dB

(b) Config. 2: QPSK, $R_c = \frac{3}{4}$, E_b/N_0=5.75 dB

(c) Config. 3: QAM64, $R_c = \frac{1}{2}$, E_b/N_0=8.5 dB

(d) Config. 4: QAM64, $R_c = \frac{2}{3}$, E_b/N_0=11.25 dB

Figure 3.26: BER in function of iterations for Butterfly and Butterfly-Replica for different configurations.

For generalization, different system configurations (for low and high modulation schemes, with different code rates) are considered: QPSK modulation with R_c=$\frac{2}{3}$, $\frac{3}{4}$ (Config. 1 and 2), and QAM64 modulation with R_c=$\frac{1}{2}$, $\frac{2}{3}$ (Config. 3 and 4). For simplicity, we consider that the SISO demapper and SISO decoder have identical symbol throughput for all the considered configurations, hence D_r which depends on the demapper and decoder architectures is equal to 1. The coded frame size is fixed to N_{CSymb}=768 symbols.

For each configuration, the number of SISO demapers and SISO decoders is chosen such that the size of the turbo decoder sub-blocks does not exceed 128 coded symbols. Applying equation (3.51) for Config. 1 leads to $\frac{J}{I}$=$\frac{2}{3}$. We choose a sub-block size of 96 coded symbols ($J = 8$) as an example. Thus, the system will be composed by $2J = 16$ SISO decoders and $I = 12$ SISO demappers. For the Config. 2, 3 and 4, sub-block sizes of 128, 64, and 96 are chosen respectively.

Fig. 3.26 illustrates the BER performance for the four configurations as a function of the number of iterations when the two schemes *B* and *B-R* are applied. The correspondence curves are plotted for a particular E_b/N_0 value as indicated in each subfigure. These E_b/N_0 values are chosen from the waterfall region. It is clearly seen from the four configurations that using the *B-R* scheme accelerates

the convergence of the iterative receiver with respect to the *B* scheme, *i.e.* less iterations are executed to achieve the same BER performance. Taking the example of Config. 1, 8 demapping iterations are required to achieve a BER=5.10^{-5} when *B-R* scheme is applied, while 10.5 iterations are required for *B*. It is worth to note that the reduction in number of iterations depends on the chosen E_b/N_0 value.

Similar behavior is seen for the other configurations. In fact, the complexity in terms of number of operations of the shuffled iterative receiver applying the two schemes is independent from the architecture and will be studied in the next subsection.

3.8.2.2 Complexity Analysis and Achieved Improvements G_4

The full shuffle mode integrating the *B-R* scheme in the SISO decoders has shown to offer a reduced number of iterations, which means a priori a reduced complexity with respect to the *B* scheme when targeting the same BER performance. On the other hand, the *B-R* scheme exhibits a higher complexity with respect to *B*. For the former scheme, an added complexity corresponding to the computations (additional arithmetic operations and read/write memory accesses) of extrinsic information are carried out in the left butterfly (Fig. 2.5). Moreover, α and β decoder state metric values are stored in the right butterfly in order to be used for the next iteration.

The main motivation behind this subsection is to analyze the impact of the turbo decoder schemes on the complexity of the iterative receiver. Therefore, an accurate evaluation of the complexity in terms of number and type of operations and memory accesses is required.

Let C_{sys}^{B} and C_{sys}^{B-R} be the receiver complexity (system-level) applying the *B* and *B-R* respectively. They can be expressed as follows:

$$\begin{aligned} C_{sys}^{B} &= C_{dem}^{-}(M).N_{MSymb} + it_B[C_{dem}^{+}(M).N_{MSymb} + C_{dec}^{B}.N_{CSymb}] \\ C_{sys}^{B-R} &= C_{dem}^{-}(M).N_{MSymb} + it_{B-R}[C_{dem}^{+}(M).N_{MSymb} + C_{dec}^{B-R}.N_{CSymb}] \end{aligned} \tag{3.52}$$

where it_x designates the number of iterations performed when using the scheme $x \in \{B, B-R\}$. C_{dec}^{x} represents the decoder complexity per coded symbol per iteration when using the scheme x.

A complexity parameter (G_4) is also defined. It represents the system-level complexity reduction for using *B-R* rather than *B*. It can be calculated using the following expression.

$$G_4 = \frac{C_{sys}^{B} - C_{sys}^{B-R}}{C_{sys}^{B}} \tag{3.53}$$

Converting the number of modulated symbols into equivalent coded symbols (equation (3.18)) and putting equation (3.52) into equation (3.53), G_4 can be written as:

$$G_4 = \frac{\alpha[it_B - it_{B-R}]C_{dem}^{+}(M) + [it_B.C_{dec}^{B} - it_{B-R}.C_{dec}^{B-R}]}{\alpha.C_{dem}^{-}(M) + it_B[\alpha C_{dem}^{+}(M) + C_{dec}^{B}]} \tag{3.54}$$

This last equation has been used to obtain individually the complexity reductions in terms of arithmetic (G_4^{Arith}) and memory access (G_4^{Mem}) operations as shown in Table 3.12.

Looking to the operations in the SISO decoder, the additional computations and access memory implied by the *B-R* scheme can be executed in parallel with the existing complexity without any additional delays. Hence, the receiver throughput improvement for using the *B-R* scheme rather than *B* can be written as equation (3.55) since the throughput of the receiver is inversely proportional to

the number of executed demapping iterations. It is worth to note that *B-R* does not provide any area overhead comparing to *B* since the extrinsic information computation unit processing in the right butterfly can be also executed in the left butterfly.

$$G_4^{thr} = \frac{it_B - it_{B-R}}{it_{B-R}} \tag{3.55}$$

Table 3.12 presents the system-level improvements for using *B-R* scheme rather than *B* for the four configurations. In all the considered cases, G_4^{Arith} presents positive values, which means a reduced number of arithmetic operations. For QPSK modulation scheme (Config. 1 and Config. 2), using the *B-R* scheme will provide a low reduction in arithmetic operations (4.4% and 5.7%). Meanwhile for higher modulation orders as for QAM64 (Config. 3 and Config. 4), more reductions up to 18.4% and 22% are shown. In fact, for high modulation orders, the SISO demapper requires a large number of arithmetic operations in comparison with low modulation schemes. Thus, executing less iterations leads to a more important reduction in overall arithmetic operations.

Regarding the memory access, Table 3.12 shows negative values G_4^{Mem} for all the considered configurations which correspond to an increased need of read and write memory accesses when using the *B-R* scheme as explained before. On the other hand, the system throughput improvement values G_4^{thr}, computed using equation (3.55), are around 33% for the four configurations.

A better investigation of the negative values in the complexity reduction of the memory access should take into consideration the values of read and write accesses apart. Table 3.13 shows the equivalent complexity of the receiver applying the two schemes in terms of number of $Add(1,1)$ operations, read and write one-bit access memory. For QPSK modulation, no significant change in the number of read memory access between the both schemes is observed. Meanwhile for QAM64, the use of *B-R* schemes has reduced the number of read memory accesses up to 14.6%.

For write memory access, negative values are obtained for all the considered configurations. Additional memory accesses are required for using *B-R*. However, this increase can be considered as small in comparison to the reduced number of $Add(1,1)$ operations and read memory accesses. Taking the example of Config. 3, 54375 $Add(1,1)$ operations and 647 one-bit read memory accesses are reduced at the expense of an additional use of 1021 one-bit write memory accesses when using the *B-R* scheme. Thus, the high difference in the magnitude of the reduced arithmetic operations and the increased memory accesses gives insights on the convenience of the *B-R* scheme to reduce the power consumption of the receiver.

Config.	Mod.	Code Rate	it_{B-R}	it_B	Parallelism		Improvement at BER			BER
					I	$2J$	G_4^{Arith}	G_4^{Mem}	G_4^{thr}	
Config. 1	QPSK	2/3	8	10.5	12	16	4.4%	-23.6%	31.3%	5.10^{-5}
Config. 2	QPSK	3/4	5	6.7	12	18	5.7%	-21.7%	34%	3.10^{-4}
Config. 3	QAM64	1/2	7	9.5	8	24	22%	-12.3%	35.7%	1.10^{-3}
Config. 4	QAM64	2/3	6	7.9	4	16	18.4%	-17.9%	31.7%	4.10^{-4}

Table 3.12: System-level: Reduction in number of arithmetic operations, memory access for using the *B-R* scheme rather than *B* for four different configurations.

3.9 Summary

In this chapter an effort is made to present an optimized adaptive system-level iterative receiver performing turbo demodulation with turbo decoding.

Config.	Butterfly Scheme			Butterfly-Replica Scheme			Improvement	
	$Add(1,1)$	Read one-bit	Write one-bit	$Add(1,1)$	Read one-bit	Write one-bit	Read	Write
Config. 1	62674	5103	2982	59912	5168	4352	-1.3%	-45.9%
Config. 2	38681	3162	1884	36466	3160	2706	$+0\%$	-43.6%
Config. 3	247057	9576	2774	192682	8176	3864	14.6%	-39.3%
Config. 4	162396	6683	2243	132459	6036	3264	9.7%	-45.5%

Table 3.13: System-level: Number of $Add(1,1)$, read and write one-bit memory access operations required in order to process one information symbol for the four configurations using *B* and *B-R* schemes

Convergence speed analysis is crucial in TBICM-ID-SSD systems in order to tune the number of iterations as much as possible when considering the practical implementation perspectives. Conducted analysis has demonstrated that omitting two turbo demodulation iterations without decreasing the total number of turbo decoding iterations leads to promising complexity reductions while keeping error rate performance almost unaltered. A maximum loss of 0.15 dB is shown for all modulation schemes and code rates in a fast-fading channel with and without erasure. In this regard, the complexity of the receiver was studied taking into account the equivalent arithmetic operations complexity and the memory accesses that should be performed. The number of normalized arithmetic operations is reduced from 15.4% for QPSK configuration to $\frac{2}{y}\%$ for QAM256 (e.g. for y=6 this gives a reduction of 32.4%). y is the number of TBICM-ID-SSD iterations. Similarly, the number of read access memory is reduced in a range between 7.9% to $\frac{2}{y}\%$.

Moreover a complexity adaptive iterative receiver performing TBICM-ID-SSD has been proposed. For low and medium constellation sizes, feedback to the SISO demapper has shown to reduce the complexity in terms of computation and access memory at the receiver side for identical error rate performances. This constitutes a very interesting result as it demonstrates the opposite of what is commonly assumed. In fact, the number of normalized arithmetic operations is reduced in a range between 28.9% and 45.9% for QPSK configuration for using TBICM-ID-SSD rather than TBICM-SSD with 6 iterations over fading channel with erasures. Similarly, the number of read/write access memory is reduced in a range between 29.8% and 47%. This complexity reduction increases significantly for higher turbo decoding iterations and reduces consequently the power consumption of the iterative receiver. On the other hand, for high modulation orders, as for QAM64 and QAM256, the TBICM-ID-SSD receiver should be configured in TBICM-SSD mode which provide less complexity for identical error rate performances. It is worth to note that for very low error rates, TBICM-ID-SSD configuration should be used as it provides more error correction in the error floor region. Fig. 3.27 summarizes the proposed use of the two modes depending on the system parameters.

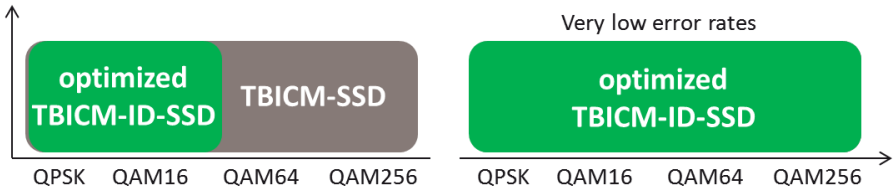


Figure 3.27: Proposal of an adaptive complexity iterative receiver applying turbo demodulation with turbo decoding.

Regarding the iterative demapping and turbo decoding area issue, this chapter proposes an approach for efficient sizing of heterogeneous multiprocessor flexible receiver processing in a serial mode. In fact, for a given communication requirement many architecture alternatives exist and selecting the right one at design-time and at run-time is an essential issue. The proposed approach defines the mathematical expressions which exhibit the number of heterogeneous cores and their features. It has been applied on a flexible multi-processor hardware platform for iterative demapping and channel

decoding. Results analysis demonstrates a reduction of the chip area of 9.6%.

Finally, this chapter has extended the use of the butterfly-replica scheme, originally proposed for shuffled turbo decoding, to full shuffled receiver implementing iterative demapping with turbo decoding. Simulation results show that applying this scheme in the turbo decoder reduces the overall number of iterations by at least one iteration in the waterfall region with respect to the butterfly scheme. In order to evaluate the impact on complexity and throughput, a detailed analysis is provided for different system configurations. Comparing butterfly-replica to the butterfly scheme for the same BER performances, the former scheme has shown to provide a throughput improvement of around 33%. Significant complexity reductions have been also obtained in terms of arithmetic operations and read access memory for almost all the considered configurations. This is particularly true for high order modulation systems. On the other hand, the number of write access memory increases, however its value remains negligible with respect to the above achieved gains. The gain in throughput and the reduced complexity were shown without any additional delays neither area overhead.

These complexity reduction techniques improve significantly latency and power consumption, and thus pave the way towards the adoption of TBICM-ID-SSD hardware implementations in future wireless receivers.

CHAPTER

4 Optimized Turbo Equalization with Turbo Decoding: Algorithms, Schedulings, and Complexity Estimation

ITERATIVE processing is being widely investigated and proposed nowadays to cope with the increasing transmission quality requirement. A typical example in this domain is the significant performance improvement offered by the use of turbo equalization with turbo decoding for applications where the propagation channel experiences multipath effects. However, these advanced techniques pose a real challenge in terms of complexity. For such iterative system, and targeting a reduced complexity MIMO turbo receiver with high error correction feature for different modulation schemes and code rates, we analyze the impact of the demapper taking into account the *a priori* information (turbo demodulation) on the MIMO system complexity.

In fact, most of the existing works have not considered the combination of multiple iterative processes from a system-level implementation point of view. The application of the iterative MIMO receiver will lead to further latency problems, more power consumption and complexity caused by the feedback inside and outside the decoder.

To the best of our knowledge, convergence speed analysis of a complete MIMO receiver using turbo equalization, turbo demodulation and turbo decoding has not been investigated in the available literature. In this chapter, we analyze the convergence speed of these combined three iterative processes in order to determine the exact required number of iterations at each level to address the ever increasing requirements of transmission quality with low complexity. An original iteration scheduling is proposed reducing one equalization iteration with maximum performance degradation of 0.04 dB. Analyzing and normalizing the computational and memory access complexity, which directly impact latency and power consumption, demonstrates the considerable gains of the proposed scheduling and the promising contributions of the proposed analysis.

The second part of this chapter demonstrates that the adoption of turbo demodulation in the context of turbo equalization combined with turbo decoding can lead to significant complexity reduction for specific system configurations. Simulations show that applying feedback to the demapper reduces the overall number of iterations (thus arithmetic operations and memory accesses) for all modulation schemes except for QPSK. Targeting the same error rate performance, results show a complexity reduction which can reach 24.5% in arithmetic operations and 31.8% in write access memory for QAM16 modulation scheme with 4×4 MIMO Spatial Multiplexing (SM).

It is worth to note that this chapter does not provide a comparison in terms of area, as the receiver is considered to perform multi-modes turbo equalization.

4.1 State of the Art

State of the art **MIMO detection techniques** can be classified into three main categories [74]: Maximum Likelihood (ML) detection [75], Sphere Decoding (SD) [76, 77], and linear filtering [78] based detection. The ML detection, although optimal, is generally avoided in practice due to its computational complexity which increases exponentially with the number of transmit antennas. The SD-based detection presents a polynomial complexity. To perform SD, first a QR decomposition [79] of channel matrix is carried out and then tree exploration is performed. This tree search is further categorized as depth-first and breadth-first methods. The depth-first has a reduced area complexity and optimal performance, but has variable throughput with SNR. In breath-first case, the most famous algorithm is the K-best in which K best nodes are visited at each level. Hence, the complexity depends on K. A large value of K results in high complexity and good performance. Finally, the use of linear filtering based solutions like Minimum Mean-Squared Error (MMSE) linear equalizer considerably reduces the computational complexity of a MIMO detector at the expense of BER performance degradation.

In order to counterbalance the performance degradation related to the use of **MMSE linear equalization, turbo equalization** has been proposed in [24] by a research group from Telecom Bretagne. In the initial contribution, the soft feedback to the SISO equalizer was provided by a **convolutional channel decoder**. Several related contributions have investigated this detection scheme for specific MIMO system configurations [26, 80]. These works have considered an additional feedback to the soft demapper and demonstrated that more than 3-5 dB gain can be obtained compared to a non-iterative MMSE [80]. In fact, iterative equalization with low complexity MMSE-based equalizer can be proposed as an alternative to the optimal ML detection.

More recently, considering simple **convolutional codes**, several contributions have been presented targeting mainly low complexity implementation aspects. In [81], a dedicated hardware architecture for SISO MMSE equalizer was proposed. Considering QPSK and 2x2 MIMO configuration, an FPGA hardware implementation of linear precoded MIMO iterative receiver has been carried out [82, 83]. In [84], a first ASIC implementation of a SISO MMSE detector is presented. The proposed architecture is based on a low complexity SISO MMSE parallel interference cancellation algorithm for systems employing iterative MIMO equalization.

Other related contributions have considered the use of **turbo codes** rather than convolutional channel codes with SISO MMSE equalization. Turbo equalization implies in this case two iterative processes: one inside the turbo decoder and one between the turbo decoder and the SISO equalizer. We can cite in this context the following main works which consider different aspects related to the iterations scheduling, parallelism analysis, hardware architectures, FPGA prototyping, and ASIC implementations:

- Boher *et al.* from Orange Labs and INSA Rennes have proposed several contributions in this context [85–89]. Regarding the iterations scheduling, three main schemes have been analysed and compared using EXIT charts and error rate performance simulations. The first scheme corresponds to the execution of multiple turbo decoding iterations for each equalization iteration. The second one increases linearly the number of turbo decoding iterations with respect to the number of equalization iterations. Hence, $k(k+1)/2$ turbo decoding iterations are applied in total for k equalizer iterations. And the third scheme, which has been finally adopted as the most efficient, applies only one turbo decoding iteration for each equalization iteration. In addition, several parallelism techniques and operations schedulings have been proposed for latency and complexity reduction of the combined two iterative processes [85–87]. Furthermore, considering a 4x4 spatial multiplexing MIMO configuration, a complete hardware architecture and corresponding FPGA prototype have been designed [88, 89].

- Jafri *et al.* from Telecom Bretagne have conducted a parallelism study and investigated the application of frame sub-blocking and shuffled decoding/detection in the context of MMSE based turbo equalization with turbo decoding [48]. Two parallelism techniques were proposed and the results have demonstrated that significant speed gain and parallelism efficiency can be attained for different MIMO configurations without degrading the error rate performance [90]. The iterations scheduling adopted here applies also one turbo decoding iteration for each equalization iteration. In addition, a flexible hardware architecture and FPGA prototype for SISO MMSE equalization have been proposed [91, 92] based on the ASIP concept. The flexibility of the designed EquASIP allows its reuse for Alamouti code [93], Golden code [94], 2x2, 3x3, or 4x4 spatially multiplexed iterative MIMO applications with modulation order up to 64-QAM.

Further contributions have investigated the use of turbo equalization considering other low complexity equalization algorithms and/or other channel codes. In this context, we can cite the works [95–97] which propose the use of low complexity SD-based algorithms for the SISO detector. The authors in [95] and [96] have considered the use of convolutional codes, while LDPC codes are used in [97]. Finally, several techniques are proposed to optimize the soft information exchange between the SISO detector and the channel decoder [98, 99].

In the scope of this thesis work we consider an iterative receiver with SISO MMSE linear equalizer combined with a turbo decoder. The analysis of the above cited works lead to define the state of the art (most efficient) scheduling of the underlined two iterative processes as to use *"one feedback to the equalizer for each turbo decoding iteration"*. However, the results obtained in the previous chapter (for turbo demodulation and turbo decoding) have demonstrated that other more efficient iteration schedulings exist for such combined iterative processes. Thus, the idea behind this work is to extend the study conducted in the previous chapter to iterative MIMO receivers. Our objective is to investigate new iteration schedulings, for a wide set of system parameters, in order to improve the convergence speed and reduce the overall complexity of such receiver combining three iterative processes: turbo decoding, turbo demodulation, and turbo equalization.

4.2 SISO Equalization Algorithm

Let us consider the system model of Fig. 4.1 which extends the system model of Fig. 3.1 by including a SISO MMSE linear equalizer. The considered receiver model applies turbo equalization in combination with turbo demodulation and turbo decoding. Regarding the considered MIMO technique, we focus only on spatial multiplexing configuration.

The received signal vector Y (made of N_r element y_l=$\{y_l^I$,$y_l^Q\}$, where $l = 1, 2, ...N_r$) can be related to the transmitted vector X (made of N_t element x_q=$\{x_q^I$,$x_q^Q\}$, where $q = 1, 2, ...N_t$) by:

$$Y = HX + W \tag{4.1}$$

where N_t and N_r are the number of transmitted and received antennas respectively. H is the channel matrix of order $N_r \times N_t$ and W is an AWGN vector of size N_r with mean 0 and covariance matrix $\sigma_w^2 I_{N_r}$, where I_{N_r} is the identity matrix.

At the receiver side the decoding is executed serially by: SISO MMSE equalizer, SISO demapper, BICM de-interleaver, process of the natural and interleaved SISO decoders, and feedback to the SISO equalizer (*TEq*) and SISO demapper (*TEq+TDem*). Using decoder *a posteriori* information as *a priori* information for the SISO equalizer and for the SISO demapper improves the error correction results. This process continues for certain number of iterations and finally decoded bits are output.

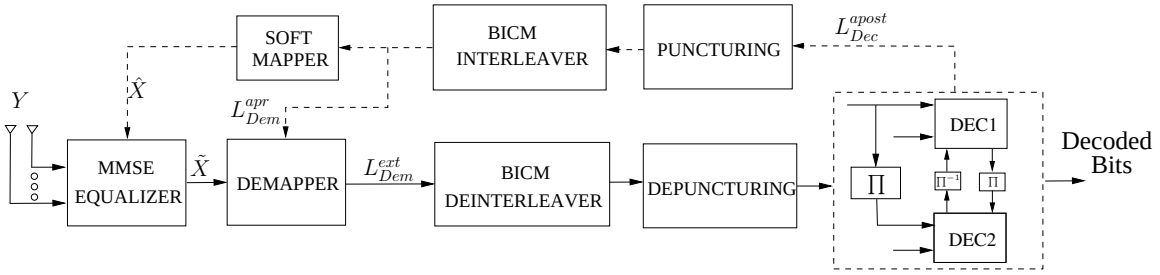


Figure 4.1: Receiver model with turbo equalization, turbo demodulation, and turbo decoding *TEq+TDem*.

4.2.1 MMSE Algorithm

The MMSE equalizer uses channel information Y vector as well as *a priori* information vector $\hat{X}$ coming from the turbo decoder to compute estimated vector $\tilde{X}$ of the transmitted vector X. After demapping, de-interleaving, de-puncturing and turbo decoding, *a posteriori* information from the turbo decoder L_{Dec}^{apost} is punctured, passed through the BICM interleaver, and fed back as *a priori* information L_{Dem}^{apr} to the demapper, and to the equalizer (after soft mapping) as *a priori* information $\hat{X}$. Note that the entire *a posteriori* information is fed back from the decoder to the equalizer and to the demapper. When considering MMSE turbo equalization of high-order modulation, it was noticed in [80] [100] a significant performance degradation when feeding back only the extrinsic information.

The equalizer's output $\tilde{X}$ is given in [101–103] as:

$$\tilde{X} = \lambda P^T (Y - H\hat{X}) + G\hat{X} \tag{4.2}$$

with

$$\lambda = \frac{\sigma_x^2}{1 + \sigma_{\hat{x}}^2 \beta} \ , \ G = \lambda\beta \ , \ \beta = diag(P^T H) \tag{4.3}$$

where λ, β and G are real positive vectors which represent the MMSE equalization coefficients. P refers to the MMSE detection vector, it can be computed as follows.

$$P = [(\sigma_x^2 - \sigma_{\hat{x}}^2) H H^T + \sigma_w^2 I_{N_r}]^{-1} H \tag{4.4}$$

where σ_x^2 and $\sigma_{\hat{x}}^2$ are the variances of transmitted and decoded symbols, $(.)^T$ is the Hermitian operator.

The variance vector σ_μ^2 of the additive distortion term taking into account the residual ISI and the filtered noise at the equalizer output can be expressed as follows.

$$\sigma_\mu^2 = G(1 - G)\sigma_x^2 \tag{4.5}$$

These expressions exhibit three main computations steps: (a) Detection vector computation referred by P (equation (4.4)), (b) Equalization coefficients computation referred by λ, β, G (equation (4.3)) and σ_μ^2 (equation (4.5)), and (c) Estimated symbols computation referred by $\tilde{X}$ (equation (4.2)).

For the first equalization iteration where no *a priori* information is presented ($\hat{X}$ is a null vector and $\sigma_{\hat{x}}$=0), the above expressions become:

$$\tilde{X} = \lambda P^T Y \tag{4.6}$$

where

$$\lambda = \sigma_x^2 \text{ and } P = [\sigma_x^2 H H^T + \sigma_w^2 I_{N_r}]^{-1} H \tag{4.7}$$

4.2.2 Demapping

In the presence of *a priori* information coming from the decoder to the demapper, equations (3.5) to (3.9) are applied. Estimated symbol $\tilde{x}_q$ corresponding to the q^{th} element of $\tilde{X}$, where $q = 1, 2, ...N_t$ is demapped by replacing $x_{r,q}$ with $\tilde{x}_q$, h'_q and h'_{q-1} by g_q (q^{th} element of G), and σ^2 by $\sigma^2_{\mu,q}$ (q^{th} element of σ^2_μ).

On the other hand, for non rotated Gray mapped constellation, and no *a priori* information coming from the decoder to the demapper, simplified equation (3.10) is used. Estimated symbol $\tilde{x}_q$ corresponding to the q^{th} element of $\tilde{X}$, where $q = 1, 2, ...N_t$ is demapped by replacing x^o_q with $\tilde{x}^o_q$, h'_q and h'_{q-1} by g_q (q^{th} element of G), and σ^2 by $\sigma^2_{\mu,q}$ (q^{th} element of σ^2_μ).

4.2.3 Soft Mapping

The LLR-to-Symbol conversion is carried out with the LLRs coming out from the BICM interleaver in the feedback loop. The estimation of $\hat{x}_q$ of transmitted symbol x_q is given by:

$$\hat{x}_q = \sum_{s \in \mathcal{X}} s \mathrm{P}\left\{x_q = s | \mathcal{L}_q\right\} \tag{4.8}$$

where $\mathcal{L}_q$ is the subset of LLRs corresponding to bits constituting the transmitted symbol x_q which is part of the normalized constellation $\mathcal{X}$ of size 2^M, M is the number of bits per modulated symbols. The term $\mathrm{P}\left\{x_q = s|\mathcal{L}_q\right\}$ designates the *a priori* probability of the symbol s. By assuming the transmitted bits statistically independent this probability becomes:

$$\mathrm{P}\left\{x_q = s|\mathcal{L}_q\right\} = \prod_{p=0}^{M-1} \mathrm{P}\left\{c_{p,q} = b\right\} \tag{4.9}$$

where $c_{p,q}$ is the p^{th} bit of symbol s_q having value $b = \{0, 1\}$ according to constellation mapping used. The term $\mathrm{P}\left\{c_{p,q} = b\right\}$, which designates the probability of $c_{p,q}$ to be equal to b, can be computed using the following equations [48].

$$\mathrm{P}\left\{c_{p,q} = 1\right\} = \frac{e^{LLR(c_{p,q})}}{1 + e^{LLR(c_{p,q})}} = \frac{1}{2}\left(1 - tanh\left(\frac{LLR(c_{p,q})}{2}\right)\right) \tag{4.10}$$

$$\mathrm{P}\left\{c_{p,q} = 0\right\} = \frac{1}{1 + e^{LLR(c_{p,q})}} = \frac{1}{2}\left(1 + tanh\left(\frac{LLR(c_{p,q})}{2}\right)\right) \tag{4.11}$$

where

$$LLR(c_{p,q}) = ln\frac{\mathrm{P}\left\{c_{p,q} = 1\right\}}{\mathrm{P}\left\{c_{p,q} = 0\right\}} \tag{4.12}$$

Taking the example of the QPSK constellation case: M=2, normalized constellation interval

$\mathcal{X}_{\mathcal{I},\mathcal{Q}}$=$\{\frac{-1}{\sqrt{2}}, \frac{+1}{\sqrt{2}}\}$, hence:

$$\hat{x}_q^I = \frac{-1}{\sqrt{2}}.\mathrm{P}\{c_{0,q} = 0\} + \frac{+1}{\sqrt{2}}.\mathrm{P}\{c_{0,q} = 1\} \tag{4.13}$$

Knowing that P$\{c_{0,q} = 0\}$+P$\{c_{0,q} = 1\}$=1, thus:

$$\hat{x}_q^I = \frac{1}{\sqrt{2}}(2.\mathrm{P}\{c_{0,q} = 1\} - 1) \tag{4.14}$$

Finally the normalized in-phase component of the estimate symbol is computed by replacing P$\{c_{0,q} = 1\}$ in equation (4.14) by its expression in equation (4.10):

$$\hat{x}_q^I = \frac{-1}{\sqrt{2}} tanh\left(\frac{LLR(c_{p,q})}{2}\right) \tag{4.15}$$

Similarly for $\hat{x}_q^Q$, it can be computed as follows.

$$\hat{x}_q^Q = \frac{-1}{\sqrt{2}} tanh\left(\frac{LLR(c_{p+1,q})}{2}\right) \tag{4.16}$$

4.2.4 Parallelism in Turbo Equalization

The proposed three level classification of parallelism techniques is extended to turbo equalization and detailed below [48].

4.2.4.1 Symbol Estimation Level Parallelism

A closer look at the expression required in MMSE algorithm with *a priori* computation (equation (4.2) to (4.5)) reveals the serial nature of the implied elementary computations. Firstly, one need to compute serially the equalization coefficients (P, β and λ) due to their related dependency and then symbols are estimated using these coefficients. The only parallelism possibility for the MMSE algorithm at this level is the temporal parallelism which can be achieved through pipelining technique.

4.2.4.2 Equalizer Component Level Parallelism

Parallelism techniques at this level can be classified in two categories: sub-block parallelism and shuffled turbo equalization.

Frame Sub-blocking: At this level, the feature which can be exploited for parallelism in the equalizer is the independence of a symbol vector from other vectors in a frame received from a memoryless channel. Hence, a linear increase in throughput can be achieved by the addition of more equalizer components to process different sub-blocks concurrently. In consequence, multiple demapper and soft mapper components will be required to balance the throughput of the multiple equalizers.

Shuffled Turbo Equalization: Once the equalizer components receive symbol vector sub-blocks, they perform symbol estimation in the absence of *priori* information. The associated demapper

components generate the LLRs from these estimated symbols in a pipelined fashion, which are de-interleaved before filling the input data memories of the decoder. After filling the decoder memories, all components of the parallel turbo equalizer work concurrently. Soft mappers, equalizers and demappers work in pipeline fashion to generate LLRs for the decoder while, on the other side, decoder components generate LLRs for the equalization side. As soon as LLRs are generated by demapper components and decoder components they are exchanged. Computation of $\sigma_{\hat{x}}^2$ is carried out during the soft mapping process and hence used in next shuffle iteration.

4.2.4.3 Turbo Equalization Level Parallelism

The highest level of parallelism duplicates the whole turbo equalization to process iterations and/or frames in parallel.

4.3 Turbo Equalization with Turbo Decoding Convergence Speed Analysis

In this section, we analyze the convergence speed of the combined three iterative processes: turbo equalization, turbo demodulation and turbo decoding. This analysis is essential in order to determine the exact required number of iterations at each level to address the ever increasing requirements of transmission quality with lower complexity.

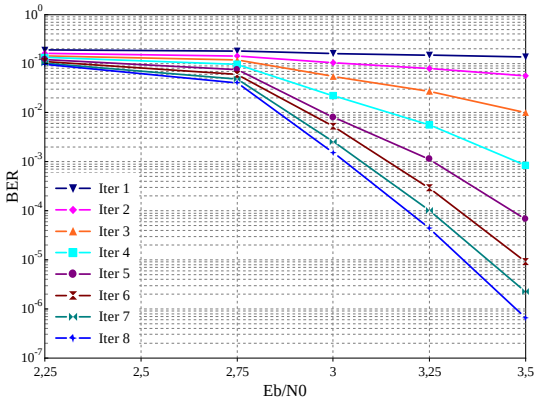
As in chapter 3, EXIT chart based analysis is conducted. For the *TEq+TDem* receiver, the *a priori* information available at the equalizer and demapper inputs improves the BER at their outputs. The resulting iterative equalization scheme is equivalent to an equalizer without *a priori* input at a higher value of $Eb/N0$. Having a changing value of $Eb/N0$ at the input of the decoder every equalizer iteration, the computation of the mutual extrinsic information IE for the turbo decoder should, as a result, also be performed per equalizer iteration.

4.3.1 *TEq* and *TEq+TDem* Error Correction Performance

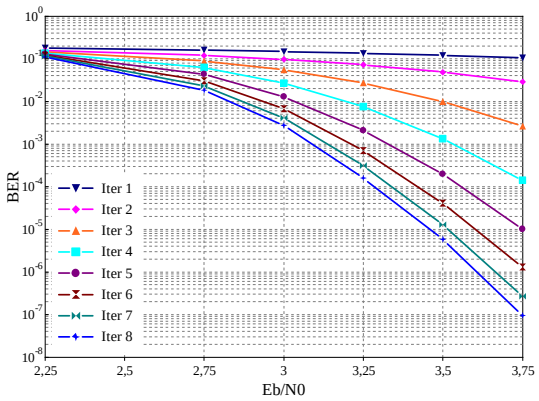
Before starting the presentation of our studies on convergence speed analysis, this sub-section gives some reference BER curves in order to compare and appreciate the error correction performance with and without feedback loop to the demapper. Fig. 4.2 presents the results of different BER simulations for *TEq* and *TEq+TDem* for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. QPSK and QAM64 modulation schemes with 2×2 and 4×4 MIMO SM and $R_c = \frac{1}{2}$ are considered. For QPSK, the 2 receiver modes offer the same BER performance since the modulated symbols are composed of two uncorrelated bits (in case of non rotated constellation). However, for all other modulation schemes, the *TEq+TDem* receiver mode provides better BER performance results than *TEq*.

4.3.2 EXIT Chart Block Diagram

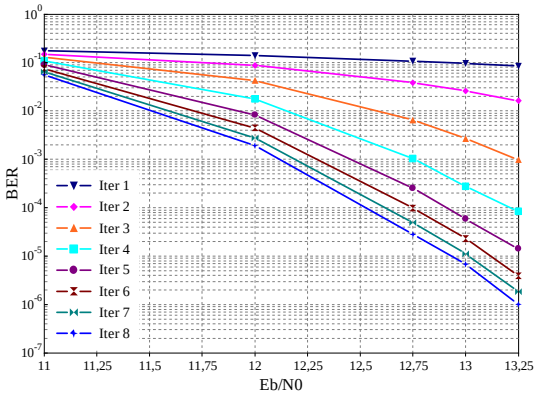
For this system receiver with three iterative processes, EXIT charts are plotted through the response of the two SISO decoders while taking into consideration the SISO equalizer and SISO demapper with updated inputs and outputs (Fig. 4.3). In this scheme, IA_1 and IE_1 are used to designate the *a priori* and extrinsic mutual information respectively for $\mathtt{DEC}_1$. Iterations start without *a priori* information ($IA_1 = 0$). Then, extrinsic information E=$\{e_1, e_2, \ldots, e_N\}$ of $\mathtt{DEC}_1$ is fed to $\mathtt{DEC}_2$ as *a priori*



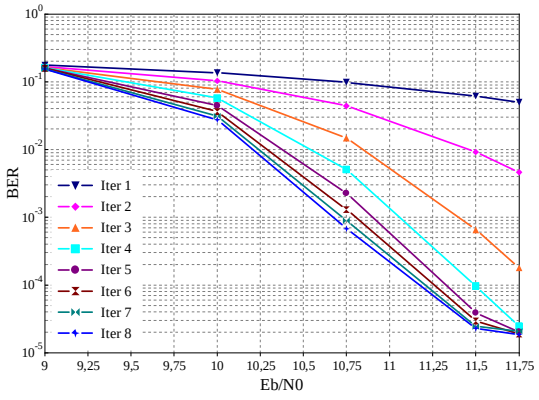
(a) *TEq* or *TEq+TDem*: QPSK with 4×4 MIMO SM



(b) *TEq* or *TEq+TDem*: QPSK with 2×2 MIMO SM



(c) *TEq+TDem*: QAM64 with 4×4 MIMO SM



(d) *TEq*: QAM64 with 2×2 MIMO SM

Figure 4.2: BER performance simulations for *TEq* and *TEq+TDem* for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. Different system configurations (QPSK and QAM64 modulation schemes with 2×2 and 4×4 MIMO SM) are considered. $R_c = \frac{1}{2}$.

information and vice versa, i.e. IE_1=IA_2 and IE_2=IA_1. N designates the number of source bits per frame. Since this EXIT chart analysis is asymptotic, long information frame size should be assumed.

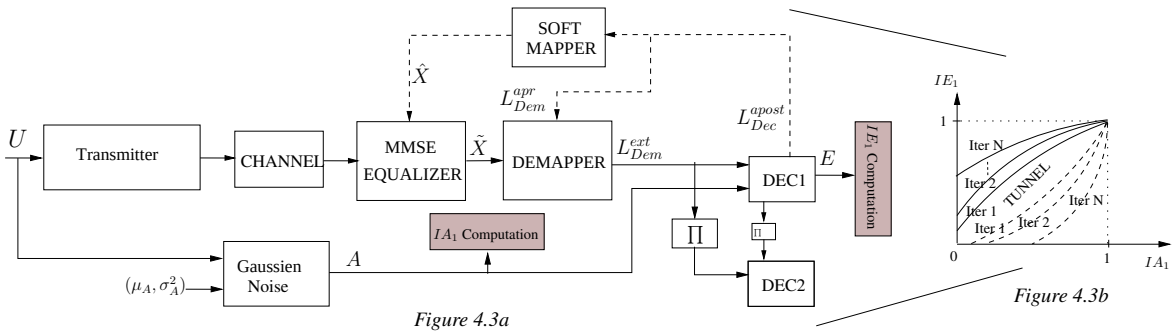


Figure 4.3a

Figure 4.3b

Figure 4.3: EXIT chart block diagram for turbo equalization combined with turbo demodulation and turbo decoding.

The transfer function of the turbo decoder is represented by the two-dimensional chart as follows (Fig. 4.3b). One SISO decoder component is plotted with its input on the horizontal axis and its output on the vertical axis. The other SISO component is plotted with its input on the vertical axis and its output on the horizontal axis. The iterative decoding corresponds to the trajectory found by stepping between the different curves. For a successful decoding, there must be a clear path between

the curves so that iterative decoding can proceed from 0 to 1 mutual extrinsic information.

4.3.3 Effects of Constellation Rotation

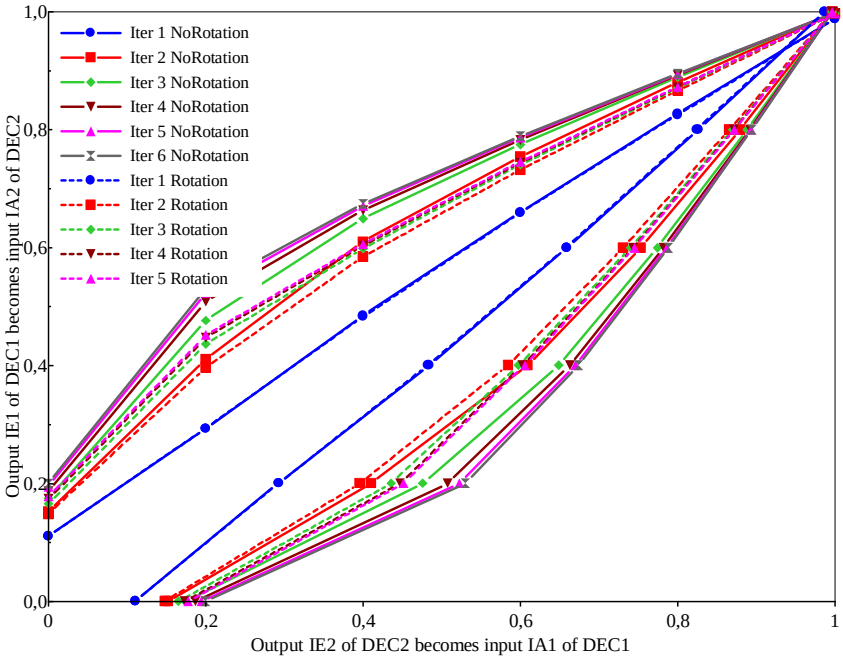


Figure 4.4: *TEq*: EXIT chart analysis at E_b/N_0= 8.25 dB of the double-binary turbo decoder for iterations to the 2×2 SM SISO MMSE equalizer. QAM16 modulation scheme and $R_c = \frac{1}{2}$ are considered for the transmission over Rayleigh fast-fading channel without erasure.

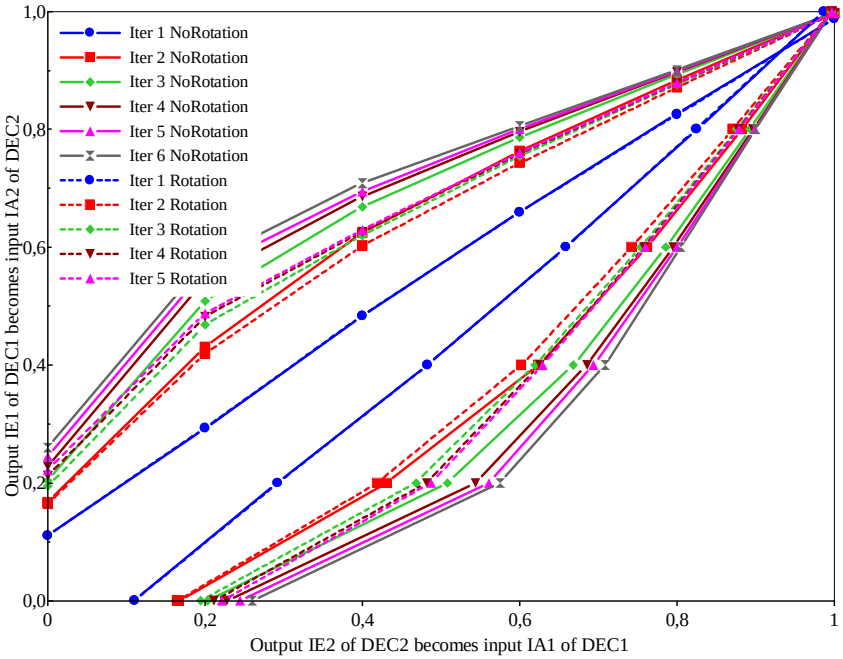


Figure 4.5: *TEq+TDem*: EXIT chart analysis at E_b/N_0= 8.25 dB of the double-binary turbo decoder for iterations to the 2×2 SM SISO MMSE equalizer and SISO demapper. QAM16 modulation scheme and $R_c = \frac{1}{2}$ are considered for the transmission over Rayleigh fast-fading channel without erasure.

Fig. 4.4 and Fig. 4.5 illustrate the effect of the constellation rotation technique on the convergence speed of the iterative MIMO receiver applying the two modes *TEq* and *TEq+TDem* respectively. 2×2 MIMO SM, QAM16, R_c=$\frac{1}{2}$, E_b/N_0=8.25 dB and P_ρ=0 are used for these two figures.

The plain curves correspond to the EXIT charts for the cases without rotated constellation. Meanwhile, the dashed curves correspond to the case with rotation. Furthermore, the blue curves correspond to non iterative equalization. Applying equalization iterations corresponds to the other colored curves in the EXIT charts of Fig. 4.4 and Fig. 4.5

In these two figures, we observe that the EXIT tunnel is smaller (except for the first iteration) for the rotated case than the one without. For rotated constellation, the tunnel is limited to that of 4 equalization iterations. Thus, making more equalization iterations will not affect the convergence speed. However the tunnel is enlarging (improving) until 5 equalization iterations for the non rotated case for the *TEq* mode and more than 5 equalization iterations for *TEq+TDem*. Hence, the non rotated constellation provides higher convergence speed comparing to the rotated case which in addition leads to a reduction in the complexity (by applying simplified expressions in subsection 4.2.2) for the two MIMO iterative receiver modes *TEq* and *TEq+TDem*.

Similar results have been found for all considered modulation orders, code rates and for 2×2 and 4×4 MIMO SM. Therefore, for the rest of this chapter, no demapper rotated constellation will be used.

4.3.4 Effects of Feedback to the Equalizer and to the Demapper

Fig. 4.6 and Fig. 4.7 illustrate respectively the EXIT charts for QAM16 with 2×2 MIMO SM and for QAM64 with 4×4 MIMO SM. R_c=$\frac{1}{2}$ and different E_b/N_0 values are considered. These values are chosen from the E_b/N_0 interval located in the waterfall region. The plain curves correspond to the EXIT charts for the *TEq* case. Meanwhile, the dashed curves correspond to the case of *TEq+TDem*. Furthermore, the blue curves correspond to non iterative equalization, i.e. SISO equalizer and SISO demapper are executed once. Applying equalization iterations corresponds to the other colored curves in the EXIT charts of Fig. 4.6 and Fig. 4.7.

In these two figures, we observe that the EXIT tunnel is wider for the *TEq+TDem* case than for the *TEq* since the demapper *a priori* information accelerates the convergence speed. Furthermore, the tunnel is enlarging (improving) until 6 equalization iterations for *TEq+TDem*, meanwhile it is blocked for 5 iterations for *TEq*. Moreover, EXIT charts of Fig. 4.6 show a need of 7 equalization iterations for *TEq* to attain convergence following the trajectory (1). Whereas 6 equalization iterations are sufficient following the trajectory (2) for *TEq+TDem*. Extensive analysis for different E_b/N_0 and different system parameters (modulation orders and code rates) has been conducted and gave identical results.

Thus, the equalization iteration scheduling which optimize the convergence is the one that enlarge the EXIT tunnel as soon as possible. Analyzing the different tunnel curves in the EXIT figures shows that the tunnel is enlarging for each equalization iteration. Hence, the optimized scheduling for the two 2 modes *TEq* and *TEq+TDem* is to execute only one turbo decoding iteration for each equalization iteration and then step forward to the next equalization iteration (enlarge the EXIT tunnel).

Note that after the fifth and sixth equalization iteration for *TEq* and *TEq+TDem* respectively, only a slight improvement in convergence is observed. This result will be used in the next section to reduce the number of equalization iterations.

Fig. 4.8 illustrates the BER performance for QAM16 modulation scheme and R_c=$\frac{1}{2}$ as a function of the number of iterations for the two schedulings *TEq* and *TEq+TDem*. This figure plots the results

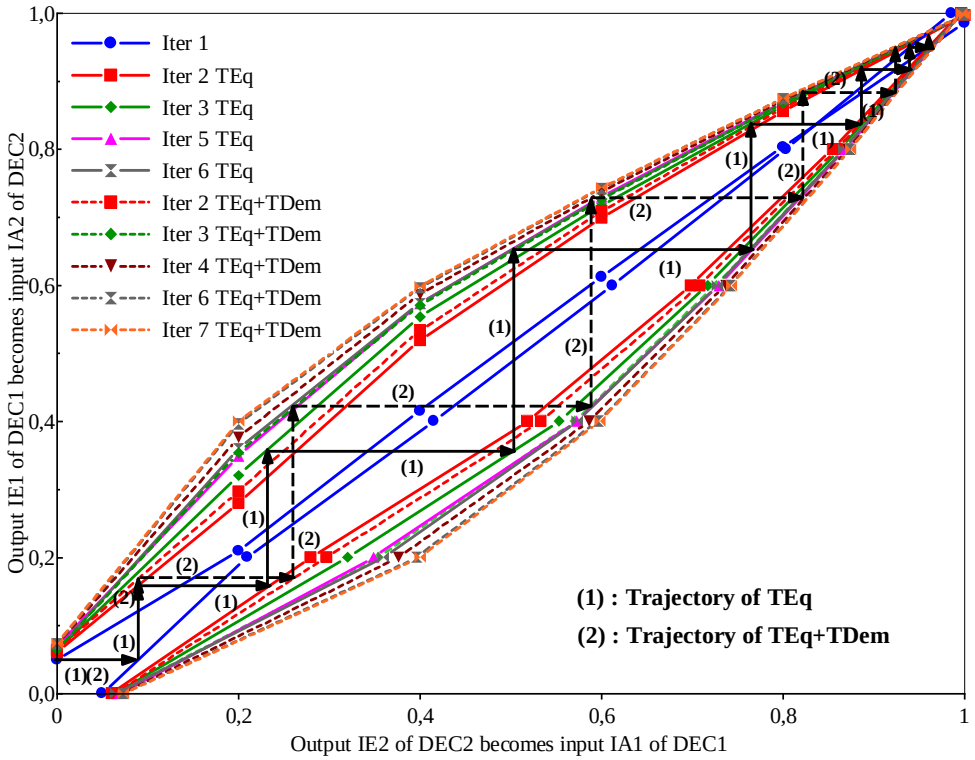


Figure 4.6: EXIT chart analysis at E_b/N_0=7.25 dB of the double-binary turbo decoder for iterations to the QAM16 demapper and MMSE equalizer. 2×2 MIMO SM and R_c=$\frac{1}{2}$ are considered for the transmission over Rayleigh fast-fading channel without erasure.

for two different E_b/N_0 values in the waterfall region and for two different MIMO configurations. These curves confirm the above EXIT chart results: using the *TEq+TDem* scheduling will accelerate the convergence of the iterative MIMO receiver with respect to *TEq*. Taking the example of the 2×2 MIMO SM case, 9 equalization iterations are required to achieve a BER=1,5.10^{-4} when *TEq* is applied, while 6 iterations are required for *TEq+TDem*. Hence, the *TEq+TDem* scheduling offers less number of iterations, which depend on the chosen E_b/N_0 value, comparing to *TEq*. Different system configurations has been simulated and gave similar results except for QPSK. For QPSK, the 2 schedulings offer the same BER performance and the same convergence speed since the modulated symbols are composed of two uncorrelated bits.

4.4 Reducing the Number of Equalization Iterations in *TEq* and *TEq+TDem*

As mentioned in the previous section, the optimized profile of iterations is the one applying one turbo decoding iteration for each equalization iteration. Thus, reducing the number of turbo equalization iterations will reduce the total number of iterations for the turbo decoder.

4.4.1 Proposed *TEq* and *TEq+TDem* Schedulings

Various constructed EXIT charts with different parameters show that after a specific number of equalization iterations, only a slight improvement is predicted. As an example, in Fig. 4.6 decoder transfer

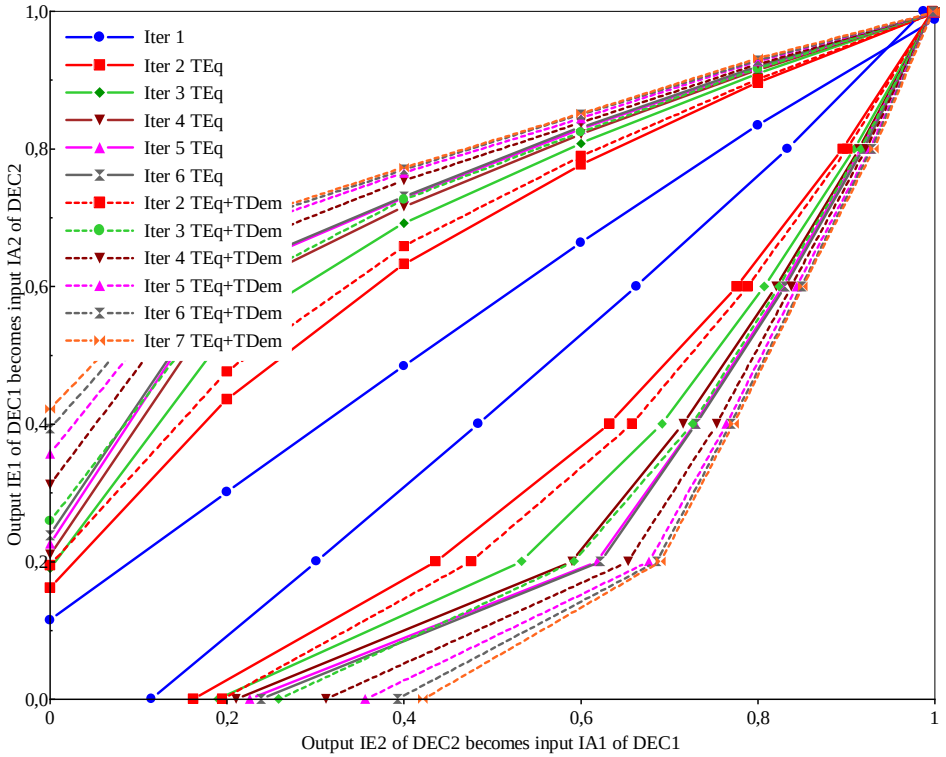


Figure 4.7: EXIT chart analysis at E_b/N_0=14 dB of the double-binary turbo decoder for iterations to the QAM64 demapper and MMSE equalizer. 4×4 MIMO SM and R_c=$\frac{1}{2}$ are considered for the transmission over Rayleigh fast-fading channel without erasure.

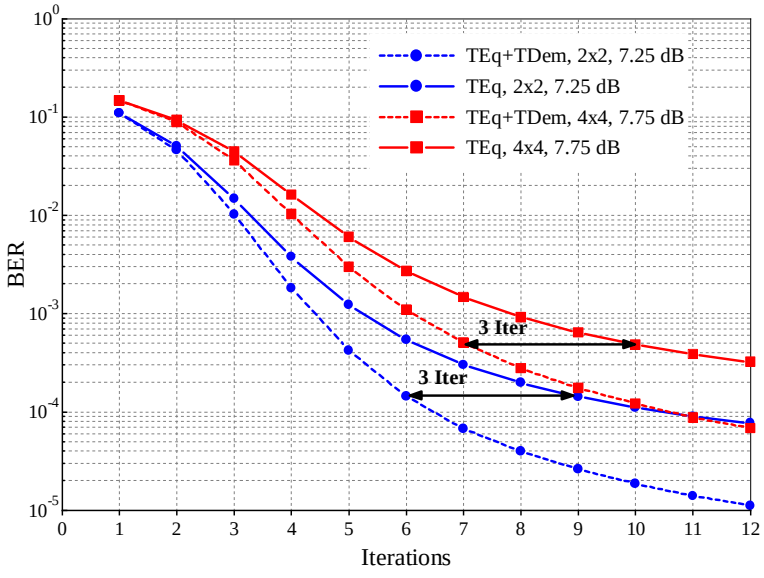
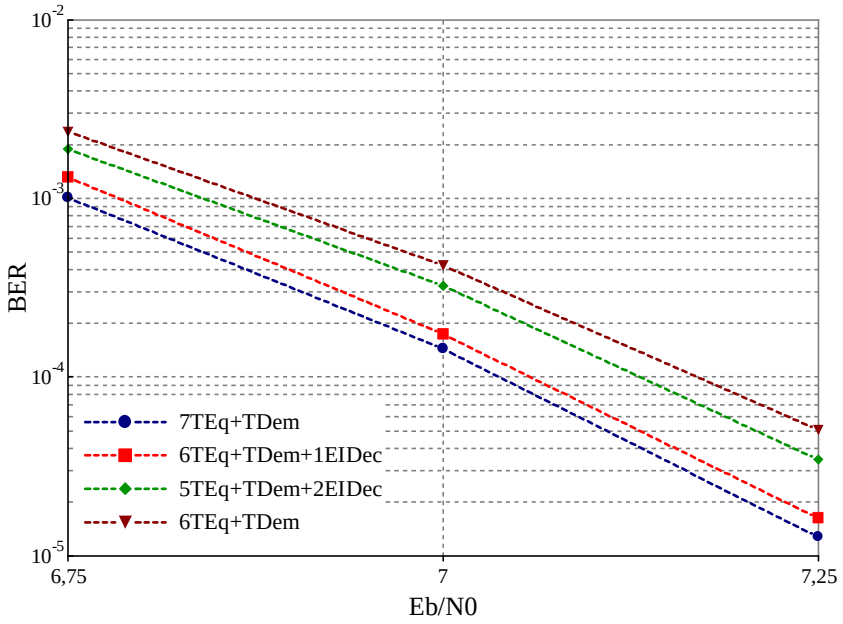
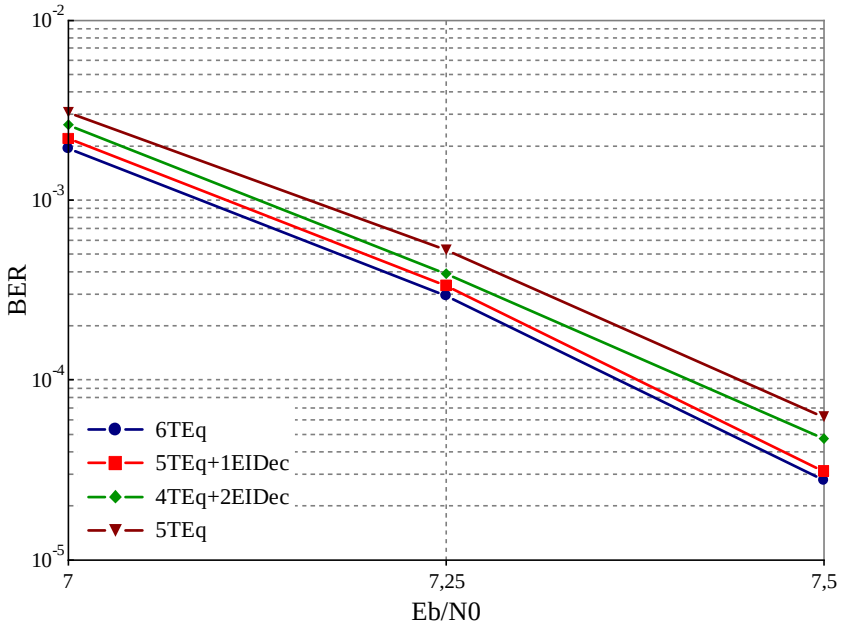


Figure 4.8: BER in function of iterations for 2×2 (E_b/N_0=7.25 dB same as for Fig. 4.6) and 4×4 (E_b/N_0=7.75 dB) MIMO SM for QAM16 modulation scheme and R_c=$\frac{1}{2}$. The coded frame size is taken as 768 double-binary symbols.

functions coincide with each other after 5 and 6 equalization iterations for *TEq* and *TEq+TDem* respectively. However, one can notice that turbo decoding iterations must continue until that the two constituent decoders agree with each other. Thus, the number of equalization iterations can be reduced without affecting error rates, while keeping the same total number of turbo decoding iterations.



(a) *TEq+TDem* modified scheduling



(b) *TEq* modified scheduling

Figure 4.9: BER performance comparison for the transmission of 1536 information bits frame over Rayleigh fast-fading channel. 2×2 MIMO SM, QAM16 modulation scheme, and code rate R_c=$\frac{1}{2}$ are considered: (a) *TEq+TDem* modified scheduling (b) *TEq* modified scheduling.

This constitutes the basis for our proposed original iteration scheduling.

In fact, to keep the same number of iterations for the decoder unaltered, one turbo decoding iteration is added after the last iteration to the equalizer for each eliminated equalization iteration. Fig. 4.9a and Fig. 4.9b simulate respectively seven turbo equalization iterations for *TEq+TDem* and six turbo equalization iterations for *TEq*. Hence, seven and six turbo decoding iterations are performed respectively.

With the proposed iteration scheduling, 6*TEq+TDem*+1*EIDec* (Fig. 4.9a) designates six equal-

ization iterations applying *TEq+TDem* (one turbo decoding iteration is applied for each) followed by one extra turbo decoding iteration.

Referring to Fig. 4.9a, error rates associated to 7*TEq+TDem* and 6*TEq+TDem+1EIDec* show almost same performances, while one feedback to the equalizer is eliminated in this proposed scheduling. Similarly, for Fig. 4.9b where error rates associated to 6*TEq* and 5*TEq*+1*EIDec* show almost same performances.

The proposed scheduling has been applied to many other system configurations: modulation order up to QAM256, different code rates, 2×2 and 4×4 MIMO SM. The maximum loss for all these configurations does not exceed 0.04 dB which corresponds to the configuration of 4×4 MIMO SM, QAM256, and R_c=$\frac{1}{2}$.

Table 4.1 summarizes the reduced performance loss for different code rates, constellation orders and number of antennas after omitting one equalization iteration. These values were investigated in the waterfall region for the worst case (minimum required number of five equalization iterations for *TEq* and 6 equalization iterations for *TEq+TDem*).

Modulation scheme	Performance loss (dB)	
	2×2 MIMO SM $R_c = 6/7 \multimap\triangleright R_c = 1/2$	4×4 MIMO SM $R_c = 6/7 \multimap\triangleright R_c = 1/2$
QPSK	$0.01 \multimap\triangleright 0.01$	$0.01 \multimap\triangleright 0.02$
QAM16	$0.01 \multimap\triangleright 0.02$	$0.01 \multimap\triangleright 0.02$
QAM64	$0.02 \multimap\triangleright 0.02$	$0.02 \multimap\triangleright 0.03$
QAM256	$0.02 \multimap\triangleright 0.04$	$0.03 \multimap\triangleright 0.04$

Table 4.1: Performance loss for different modulation schemes, code rates, and number of antennas after one omitted equalization iteration over Rayleigh fast-fading channel without erasure.

For 4*TEq*+2*EIDec* and 5*TEq+TDem*+2*EIDec*, two feedbacks to the equalizer are eliminated. An additional loss (maximum loss of 0.15 dB for all possible configurations) is seen. These curves are close to 5*TEq* and 6*TEq+TDem* respectively than to 6*TEq* and 5*TEq*+1*EIDec*. Hence, eliminating two feedbacks does not provide an optimized solution for the original *TEq* and *TEq+TDem* schedulings. Eliminating more equalization iterations will lead to significant BER performance degradation.

4.4.2 SISO MMSE Equalizer Complexity Evaluation

The main motivation behind the conducted convergence speed analysis and the proposed technique for reducing the number of iterations for *TEq* and *TEq+TDem* is to reduce the receiver implementation complexity. In order to appreciate the achieved improvements, an accurate evaluation of the complexity in terms of number and type of operations and memory access is required. Such complexity evaluation is fair and generalized as it is independent from the architecture mode (serial or parallel) and remains valid for both of them. In fact, all architecture alternatives should execute the same number of operations (serially or concurrently) to process a received frame. To this end, The two main blocks of section 3.4.2 which are the SISO demapper and the SISO decoder are considered. Complexity evaluation of the SISO MMSE equalizer is also required. For this latter, the proposed evaluation considers the low complexity algorithm presented in section 4.2.1.

4.4.2.1 SISO Equalization Typical Quantization Values

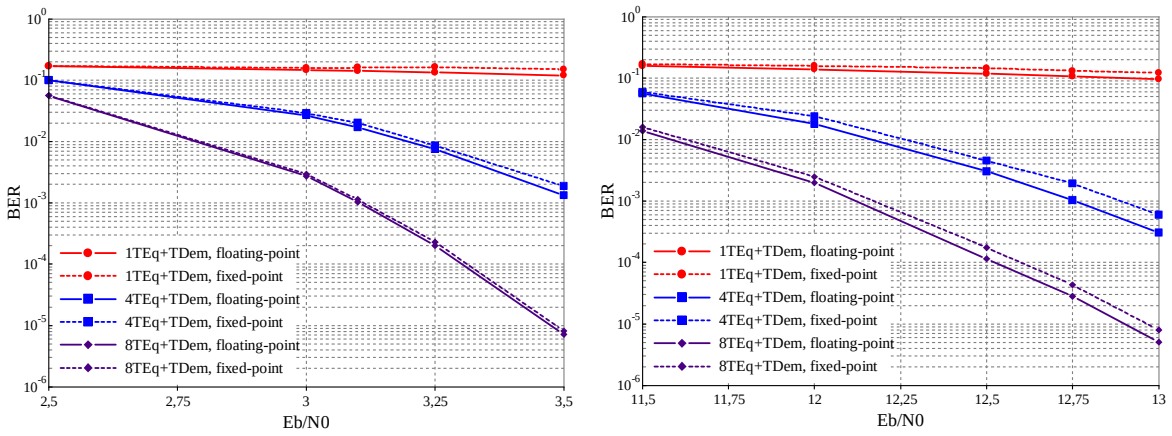
A typical fixed-point representation of channel inputs and various metrics is considered. Table 4.2 summarizes the total number of required quantization bits for each parameter of the MMSE SISO

equalizer [48].

	Parameter	Number of bits
SISO equalizer	Received complex symbol y_l=$\{y_l^I, y_l^Q\}$	{12,12}
	Complex Coeff. Fading symbol $h_{q,l}$=$\{h_{q,l}^I, h_{q,l}^Q\}$	{12,12}
	Estimated complex symbol $\tilde{x}_q$= $\{\tilde{x}_q^I, \tilde{x}_q^Q\}$	{16,16}
	A priori information complex symbol $\hat{x}_q$= $\{\hat{x}_q^I, \hat{x}_q^Q\}$	{16,16}
	Detection complex symbol p_q=$\{p_q^I$,$p_q^Q\}$	{16,16}
	Bias symbol g_q	16
	Distortion variance symbol $\sigma_{\mu,q}^2$	16
	Equalization coefficient symbol λ_q	16
	Equalization coefficient symbol β_q	16

Table 4.2: MMSE SISO equalization typical quantization values.

Using this quantization, Fig. 4.10 plots two sets of floating-point vs fixed-point BER performance curves for *TEq+TDem*. Two modulation schemes, QPSK and QAM64, and different number of iterations are considered with R_c=$\frac{1}{2}$. As we can see from this figure, considering the quantization of Table 4.2 provides almost the same BER as for the floating-point reference performance.



(a) QPSK, 2×2 MIMO SM **(b)** QAM64, 4×4 MIMO SM

Figure 4.10: Floating-point vs Fixed-point BER performance comparison for *TEq+TDem* for the transmission of 1536 information bits frame over Rayleigh fast-fading channel with erasure. QPSK with 2×2 MIMO SM and QAM64 with 4×4 MIMO SM are considered respectively for 1, 4, and 8 iterations. R_c=$\frac{1}{2}$.

4.4.2.2 Complexity Evaluation of SISO Equalizer

The complexity of MMSE-SISO equalizer depends on the number of antennas. We will now consider the equations of subsection 4.2.1 to compute: (1) the required number and type of arithmetic computations and (2) the required number of read memory accesses ($load$) and write memory accesses ($store$). The result of this evaluation is explained below. In addition to the operation notations used in section 3.4.2.2, we use the notation $operation^o$ to designate an operation having one of the operands, or the resulted operand, as a matrix with real diagonal values. Semi-complex operation is also used to designate an operation between a complex number/matrix and a real number/matrix. The Rayleigh fast-fading channel is considered, hence the channel fading vector H will be different for each received complex vector Y.

		Number and Type of operations per received vector per turbo equalization iteration	
SISO MMSE equalizer with *a priori* input	Computation units	2×2 MIMO SM	4×4 MIMO SM
	Detection vector	10146$Add(1,1)$ + $load(193)$	111017$Add(1,1)$ + $load(482)$
	Equalization coefficients	5235$Add(1,1)$	16578$Add(1,1)$
	Estimated symbol	9913$Add(1,1)$ + $load(128)$ + $store(64)$	38030$Add(1,1)$ + $load(256)$ + $store(128)$

Table 4.3: MMSE-SISO equalization complexity computation summary with *a priori* information after normalization.

		Number and Type of operations per received vector per turbo equalization iteration	
SISO MMSE equalizer without *a priori* input	Computation units	2×2 MIMO SM	4×4 MIMO SM
	Detection vector	10130$Add(1,1)$ + $load(161)$	111001$Add(1,1)$ + $load(450)$
	Equalization coefficients	4332$Add(1,1)$	14772$Add(1,1)$
	Estimated symbol	5986$Add(1,1)$ + $load(64)$ + $store(64)$	24068$Add(1,1)$ + $load(128)$ + $store(128)$

Table 4.4: MMSE-SISO equalization complexity computation summary without *a priori* information after normalization.

1) Detection vector computation P (equation (4.4))
For each received vector Y (input of the equalizer):

- 3 $load(16)$ to access the σ_x^2, $\sigma_{\hat{x}}^2$ and σ_w^2 variances.
- 1 real $Sub(16,16)$ to compute $(\sigma_x^2 - \sigma_{\hat{x}}^2)$
- 1 $load$ of complex matrix H
- 1 hermitian operation to compute H^T
- 1 complex matrix Mul^o to compute HH^T. The obtained matrix has its diagonal as real values.
- 1 semi-complex matrix Mul^o to multiply the resulted matrix above with $(\sigma_x^2 - \sigma_{\hat{x}}^2)$
- 1 semi-complex matrix Add^o to add the result above with $\sigma_w^2 I_{N_r}$
- 1 complex matrix Inv^o to make the inversion of the matrix above.
- 1 complex matrix Mul^o to multiply the matrix above with the H matrix (computing P)

2) Equalization coefficients computation λ, β, G and σ_μ^2 (equation (4.3))
For each received vector Y (input of the equalizer):

- 1 hermitian operation to compute P^T
- 1 complex matrix Mul^o to multiply the resulted matrix above with H and taking the diagonal which is made by real values (computing β)
- 1 real matrix Mul to multiply β with $\sigma_{\hat{x}}^2$
- 1 real matrix Add to add 1 to the resulted matrix above
- 1 real matrix Inv to make the inversion of the matrix above
- 1 real matrix Mul to multiply the matrix above with σ_x^2 (computing λ)
- 1 real matrix Mul to multiply λ with β (computing G)
- 1 real matrix Sub to compute (1-G)
- 1 real matrix Mul to multiply the matrix above with matrix G

- 1 real matrix Mul to multiply the matrix above with σ_x^2 (computing σ_μ^2)
- 1 $store$ of real matrix σ_μ^2

3) Estimated symbols computation $\tilde{X}$ (equation (4.2))
For each received vector Y (input of the equalizer):

- 2 $load$ of complex vectors $\hat{X}$ and Y
- 1 complex matrix Mul to multiply the matrix H with $\hat{X}$
- 1 complex matrix Sub to compute $(Y - H\hat{X})$
- 1 semi-complex matrix Mul to multiply P^T with λ
- 1 complex matrix Mul to multiply the two resulted matrices above
- 1 semi-complex Mul to compute $G\hat{X}$
- 1 complex matrix Add to add the two resulted matrices above (computing $\tilde{X}$)
- 1 $store$ of complex matrix $\tilde{X}$

Note that $\sigma_{\hat{x}}^2$ should be computed, for each equalization iteration, after the soft mapping of all decoded symbols. For the rest of this chapter, we will consider identical number of antennas N_r=N_t=2 or 4.

4.4.2.3 Complexity Normalization

In addition to the complexity normalization technique used in subsection 3.4.2.4, we use the following complex operations normalization approach.

1. **Complex number operations**

   - Add/Sub of two complex numbers X=$(a+jb)$ and Y=$(c+jd)$ can be performed with two real Add/Sub operations.

   $$X \pm Y = (a + jb) \pm (c + jd) = (a \pm c) + j(b \pm d) \tag{4.17}$$

   - Negation of a complex number Neg can be performed with two real Sub operations.
   - Mul of two complex numbers X=$(a+jb)$ and Y=$(c+jd)$ can be performed by two ways. The classical formula of equation (4.18) performs 4 real Mul, 1 real Add and 1 real Sub.

   $$X \times Y = (a + jb)(c + jd) = (ac - bd) + j(ad + bc) \tag{4.18}$$

   A rearrangement may be proposed to reduce the number of multiplications required, as:

   $$X \times Y = (a + jb)(c + jd) = a(c + d) - d(a + b) + j\,[a(c + d) + c(b - a)] \tag{4.19}$$

   By applying this reformulation, a complex number multiplication must perform only 3 real Mul, 3 real Add and 2 real Sub. Reducing one real multiplication operation per complex multiplication operation at the cost of three additional addition operations significantly reduces the complexity of the complex number multiplication. For the rest of this normalization section, equation (4.19) will be used.

- Inv of a complex number $a + bj$ can be performed with two read memory access to Look Up Table (LUT). In fact, the inverse of $a+bj$ can be computed using the following expression.

$$\frac{1}{a+bj} = \frac{a}{a^2+b^2} - \frac{b}{a^2+b^2}j \tag{4.20}$$

  Hence two $load$ of $\frac{a}{a^2+b^2}$ and $\frac{-b}{a^2+b^2}$ are required

2. **Complex matrix operations**

- Hermitian operation of complex matrix of size $b\times a$ can be performed with ab negation operations of the imaginary part.
- Add/Sub of two complex matrices of size $b\times a$ can be performed with ab complex Add/Sub operations which correspond to $2ab$ real Add/Sub operations.
- Mul of two complex matrices of size $b\times a$ and $a\times c$ can be performed with abc complex Mul and $(a-1)bc$ complex Add operations. Using equation (4.19), this matrix multiplication can be done with $3abc$ real Mul, $(5a-2)$bc real Add and $2abc$ real Sub operations.
- Mul of two complex matrices of size $b\times a$ and $a\times c$ with resulted matrix having real diagonal values can be performed with $3abc-ab$ real Mul, $5abc-4ab-2bc+b$ real Add and $2abc-ab$ real Sub operations.
- Mul of two complex matrices of the same size $a\times a$ where one of the matrices has real diagonal values can be performed with $3a^3-2a^2$ real Mul, $5(a^3-a^2)$ real Add and $2(a^3-a^2)$ real Sub.
- Mul of the identity matrix having real diagonal values with complex matrix of size $b\times a$ can be performed with ab complex Mul which correspond to $3ab$ real Mul, $3ab$ real Add and $2ab$ real Sub.
- Inv of a complex matrix can be achieved through matrix triangulation or analytical method. The first method based on matrix triangulation can realized using systolic architecture through the LU decomposition, Cholesky decomposition or QR decomposition. The method based on QR decomposition is the most interesting due to its numerical stability and its practical feasibility. It consists of decomposing a matrix $\mathbf{A}$ of size $N \times N$ as $\mathbf{A} = \mathbf{QR}$ where $\mathbf{Q}$ is an orthogonal matrix ($\mathbf{QQ}^H = \mathbf{I}$) and $\mathbf{R}$ an upper triangular matrix. This decomposition allows to compute the inverse of the matrix $\mathbf{A}$ after a simple inversion of the triangular matrix $\mathbf{R}$ and a matrix multiplication as $\mathbf{A}^{-1} = \mathbf{R}^{-1}\mathbf{Q}$. There are several methods [104] to achieve this decomposition, such as the Givens method or the method of Gram-Schmidt. On the oher hand, the analytic method of matrix inversion is good candidate, not only for variable sized matrix inversion but also for resource reuse for other matrix computations. The expression for the inversion of 2×2 matrix through analytical method is given by:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{4.21}$$

  To compute the inversion matrix of expression (4.21), 1 Inv of $ad-bc$, 2 Neg and 4 Mul are required. For a 4×4 matrix, the matrix is divided into four 2×2 matrix and inversion can be achieved blockwise.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} W & X \\ Y & Z \end{bmatrix} \tag{4.22}$$

where

$$\begin{aligned} W &= A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \\ X &= -A^{-1}B(D - CA^{-1}B)^{-1} \\ Y &= -(D - CA^{-1}B)^{-1}CA^{-1} \\ Z &= (D - CA^{-1}B)^{-1} \end{aligned}$$

The inversion of a 3×3 matrix is performed by extending it to a 4×4 matrix. This can be done by copying all three rows of 3×3 matrix into first three rows of 4×4 matrix and then putting zeros in all elements of fourth row and fourth column where a 1 should be put on the intersection of fourth row and fourth column. The inversion can then be performed using the method mentioned above. The final result lies in first three elements of first three rows (or column).

Applying the proposed complexity normalization approach presented in this subsection to the complexity evaluation of the MMSE-SISO equalizer with and without *a priori* information (subsection 4.4.2.2) leads to the results summarized in Table 4.3 and Table 4.4 respectively.

4.4.3 Discussions and Achieved Improvements

This section evaluates and discusses the achieved complexity reductions using the proposed original iteration scheduling of *TEq* and *TEq+TDem* at different modulation orders, code rates and number of antennas. As concluded in section 4.4.1, one equalization iteration can be eliminated while keeping the number of turbo decoding iterations unaltered. Overall, this will lead to a reduction corresponding to the execution of one SISO MMSE equalizer for *TEq* and one SISO MMSE combined with one SISO demapper for *TEq+TDem*. The low complexity sub-partitioning technique (subsection 1.1.3.4) of the constellation presented in [21] for non constellation rotated case is used. Using the normalized complexity evaluation of Table 4.3 and Table 4.4, achieved improvements comparing (1): $7TEq$+$TDem$ to $6TEq$+$TDem$+$1EIDec$ and (2): $6TEq$ to $5TEq$+$1EIDec$ (as in Fig. 4.9) for different system configurations are summarized in Tables 4.5 and 4.6. In the following we will explain first how these values are computed and then discuss the obtained results.

4.4.3.1 Complexity Reduction Ratio G_5

The complexity reduction ratio (G_5) is defined as the ratio of the difference in complexity (system-level) between the original scheduling (C_{sch}^{sys}), $sch \in \{TEq, TEq + TDem\}$, and the new proposed scheduling ($C_{NEW-sch}^{sys}$) to the complexity C_{sch}^{sys} for processing all frame source symbols. It corresponds to the complexity reduction ratio when using the proposed scheduling *NEW-sch*. G_5 can be expressed as follows:

$$G_5 = \frac{C_{sch}^{sys} - C_{NEW-sch}^{sys}}{C_{sch}^{sys}} \tag{4.23}$$

where C_{sch}^{sys} and $C_{NEW-sch}^{sys}$ can be expressed as below.

$$C_{sch}^{sys} = C + [it_{sch} - 1]C_{sch} \tag{4.24}$$

$$C_{NEW-sch}^{sys} = C + [(it_{sch} - 1) - 1]C_{sch} + C_D \tag{4.25}$$

C (same for C_{sch}^{sys} and $C_{NEW-sch}^{sys}$) designates the complexity of the first iteration without taking into consideration the *a priori* information. it_{sch} and C_{sch} designate respectively the number of iterations and the complexity per iteration performed when using the corresponding scheduling sch. C_D (C_D=$C_{dec}.N_{CSymb}$) designates the complexity of executing one turbo decoding iteration. C and C_{sch} are given by the expressions below:

$$\begin{aligned} C &= C_{eq}^{-}(N_r).N_{ESymb} + C_{dem}^{-}(M)N_{MSymb} + C_{dec}.N_{CSymb} \\ C_{sch} &= C_{eq}^{+}(N_r).N_{ESymb} + [C_{dem,sch}(M) + C_{mod}(M)]N_{MSymb} + C_{dec}.N_{CSymb} \end{aligned} \quad (4.26)$$

$C_{eq}^{-}(N_r)$ designates the equalizer complexity per equalized vector per iteration, which depends only on N_r, without taking into consideration the *a priori* information $\hat{X}$. $C_{eq}^{+}(N_r)$ designates the equalizer complexity per equalized vector per iteration, which depends only on N_r, taking into consideration the *a priori* information $\hat{X}$. $C_{dem}^{-}(M)$ designates the demapper complexity per modulated symbol per iteration, which depends only on the constellation size, without taking into consideration the *a priori* information L_{Dem}^{apr}. $C_{dem,sch}(M)$ designates the demapper complexity per modulated symbol per iteration, which depends only on the constellation size, when using the corresponding scheduling sch. $C_{mod}(M)$ designates the soft mapper complexity per modulated symbol per iteration. C_{dec} represents the decoder complexity per coded symbol per iteration.

Considering the code rate R_c, the number of receive antennas N_r and the number of bits per modulated symbol M, the relations between the number of equalized symbols (N_{ESymb}) and the number of modulated symbols (N_{MSymb}) with the corresponding number of double-binary (∇=2) coded symbols (N_{CSymb}) can be written as follows.

$$\begin{aligned} N_{MSymb} &= \frac{\nabla.N_{CSymb}}{MR_c} = \alpha N_{CSymb} \text{ where } \alpha = \frac{\nabla}{MR_c} \\ N_{ESymb} &= \frac{N_{MSymb}}{N_r} = \frac{\alpha N_{CSymb}}{N_r} \end{aligned} \quad (4.27)$$

Converting the number of equalized and modulated symbols into equivalent number of coded symbols (equation (4.27)) and putting equations (4.24), (4.25) and (4.26) into equation (4.23), G_5 can be written as:

$$G_5 = \frac{a}{b} \quad (4.28)$$

where

$$a = \frac{1}{N_r} C_{eq}^{+}(N_r) + C_{dem,sch}(M) + C_{mod}(M) \quad (4.29)$$

$$b = \frac{1}{\alpha} C_{sch}^{sys} \quad (4.30)$$

a designates the reduction in complexity for using the proposed scheduling *NEW-sch*. b is equal to C_{sch}^{sys} divided by α.

4.4.3.2 Achieved Improvements

This last equation has been used to obtain individually the complexity reductions in terms of arithmetic, read memory access, and write memory access operations of Tables 4.5 and 4.6 comparing $7TEq$+$TDem$ to $6TEq$+$TDem$+$1EIDec$ and $6TEq$ to $5TEq$+$1EIDec$.

For the *TEq+TDem* case, results from Table 4.5 show increased benefits in terms of number of arithmetic operations (up to 18.7%) and read memory accesses (up to 16.6%) with higher modulation orders. This can be easily predicted from equation (4.28) as the value of a (equation (4.29)) is increasing with the constellation size faster than b (equation (4.30)). This is due to the presence of different complexity values in b other than $C_{dem,sch}(M)$ (not presented in a). These tables show also that the higher the code rate is, lower the benefits are. On the other hand, the improvement in write memory access (up to 7.3%) is low and decreasing with the constellation size.

Similar behavior is shown for the *TEq* case (Table 4.6). However, the benefits for *TEq* are less than *TEq+TDem* since one additional demapping execution process is omitted for this latter. It is worth to note in the two tables that the reduction in write memory accesses is independent from the number of antennas and shows identical values for Nr=2 and N_r=4 since the number of bits to be stored after the proposed scheduling is the same for the two cases.

Modulation scheme	*TEq+TDem*, N_r=2						*TEq+TDem*, N_r=4					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	Complexity Reduction			Complexity Reduction			Complexity Reduction			Complexity Reduction		
	$arith$	$load$	$store$	$arith$	$load$	$store$	$arith$	$load$	$store$	$arith$	$load$	$store$
QPSK	16.4%	13.5%	7.3%	15.7%	11.4%	5.6%	16.4%	13.6%	7.3%	16.2%	11.6%	5.6%
QAM16	16.7%	13.6%	5.7%	15.9%	11.5%	4.9%	16.6%	13.7%	5.7%	16.3%	12%	4.9%
QAM64	17.3%	13.9%	5%	16.3%	11.8%	4.5%	16.9%	13.9%	5%	16.4%	12.5%	4.5%
QAM256	18.7%	16.6%	4.6%	18.1%	14.8%	4.3%	18.1%	16.6%	4.6%	17.7%	14.8%	4.3%

Table 4.5: Reduction in number of operations, read/write access memory comparing $7TEq$+$TDem$ to $6TEq$+$TDem$+$1EIDec$ for 2×2 and 4×4 MIMO SM for different modulation schemes and code rates.

Modulation scheme	*TEq*, N_r=2						*TEq*, N_r=4					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	Complexity Reduction			Complexity Reduction			Complexity Reduction			Complexity Reduction		
	$arith$	$load$	$store$	$arith$	$load$	$store$	$arith$	$load$	$store$	$arith$	$load$	$store$
QPSK	14.5%	9.7%	6%	13.5%	8.3%	4.9%	14.8%	10.4%	6%	13.9%	8.6%	4.9%
QAM16	14.9%	9.9%	5%	14.1%	8.5%	4.4%	15.6%	10.6%	5%	14.4%	8.7%	4.4%
QAM64	15.8%	10.5%	4.5%	15.2%	8.9%	4%	16.7%	11%	4.5%	15.2%	9%	4%
QAM256	17.3%	13.1%	4.2%	16.4%	10.3%	3.8%	17%	11.8%	4.2%	16.5%	9.8%	3.8%

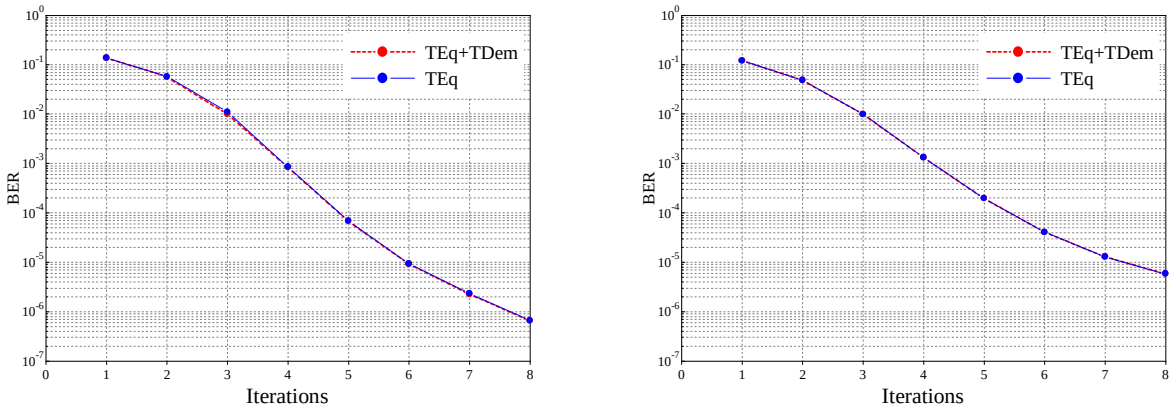
Table 4.6: Reduction in number of operations, read/write access memory comparing $6TEq$ to $5TEq$+$1EIDec$ for 2×2 and 4×4 MIMO SM for different modulation schemes and code rates.

It is worth noting that applying the proposed scheduling combined with an early stopping criteria might diminish the benefit from the scheduling, but at the cost of an additional complexity.

4.5 Complexity Adaptive *TEq+TDem* Receiver

In this section, we consider a MIMO turbo receiver which is able to execute *TEq+TDem*. Such a receiver can obviously execute the *TEq* mode by omitting the feedback loop to the demapper. In this context, the following work will discuss the complexity and the BER performance of the two modes for different configurations in terms of modulation orders and number of antennas. Based on this analysis, a complexity adaptive iterative MIMO receiver is proposed.

The conducted analysis can be done either on the original *TEq+TDem* and *TEq* schedulings or on the new proposed schedulings in the previous section. However, for presentation simplicity we have chosen the original schedulings without omitting a feedback to the last iterations.



(a) QPSK with 4×4 MIMO SM, E_b/N_0=3.5 dB: *Config. 1*

(b) QPSK with 2×2 MIMO SM, E_b/N_0=3.5 dB: *Config. 2*

(c) QAM16 with 4×4 MIMO SM, E_b/N_0=8.25 dB: *Config. 3*

(d) QAM16 with 2×2 MIMO SM, E_b/N_0=7.5 dB: *Config. 4*

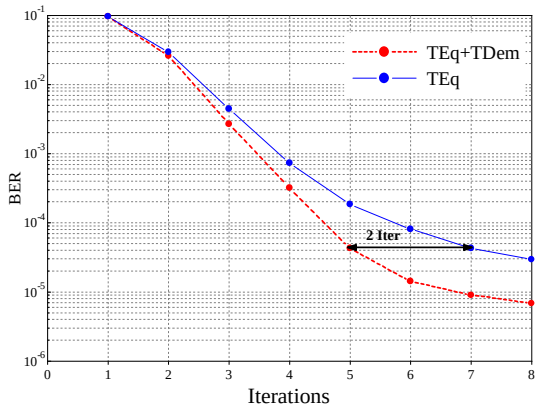
Figure 4.11: BER in function of iterations for 2×2 and 4×4 MIMO SM for QPSK and QAM16 modulation schemes. E_b/N_0 values are chosen from the E_b/N_0 interval located in the waterfall region. Rayleigh Fast-fading channel, R_c=$\frac{1}{2}$ and N_{CSymb}=768 are considered.

4.5.1 *TEq+TDem* and *TEq* Performance Simulations

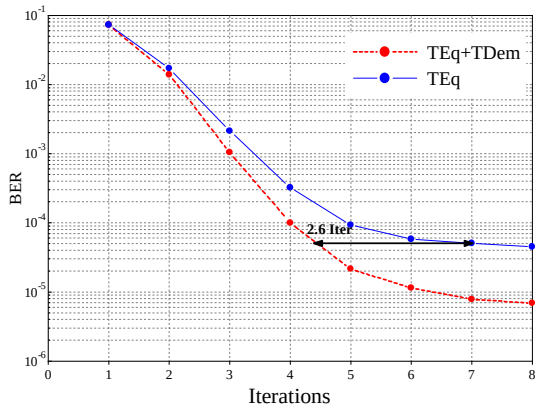
We compare in this subsection the BER performance results of the MIMO turbo receiver applying *TEq+TDem* and *TEq* for different modulation scheme and number of antennas.

A flexible software model for the whole serial system was developed. It supports different modulation schemes (QPSK, QAM16, QAM64 and QAM256) applying the sub-partitioning technique (subsection 1.1.3.4), and a flag choice for iterative or non iterative feedback loop to the demapper. A Rayleigh fast-fading channel is considered.

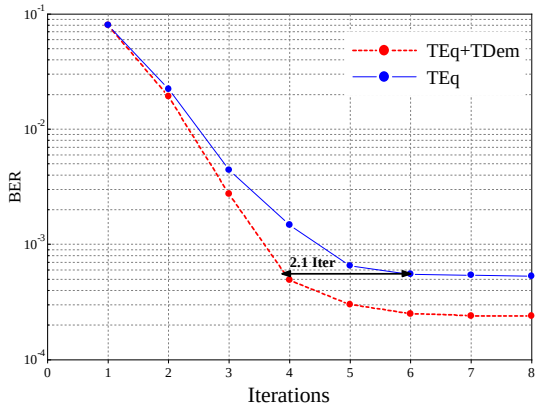
Fig. 4.11 and Fig. 4.12 illustrate the BER performance for different system configurations (for low and high modulation schemes, with 2 × 2 and 4×4 MIMO SM) as a function of the number of iterations when the two modes *TEq* and *TEq+TDem* are applied. The coded frame size is taken N_{CSymb}=768 double-binary symbols. The code rate R_c=$\frac{1}{2}$ is considered for all configurations. The



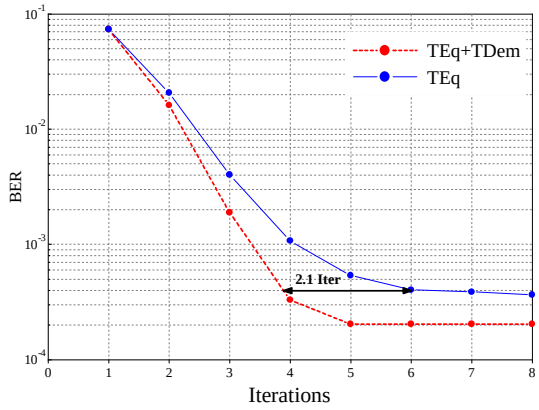
(a) QAM64 with 4×4 MIMO SM, E_b/N_0=13 dB: *Config. 5*



(b) QAM64 with 2×2 MIMO SM, E_b/N_0=11.25 dB: *Config. 6*



(c) QAM256 with 4×4 MIMO SM, E_b/N_0= 17 dB: *Config. 7*



(d) QAM256 with 2×2 MIMO SM, E_b/N_0=: *Config. 8*

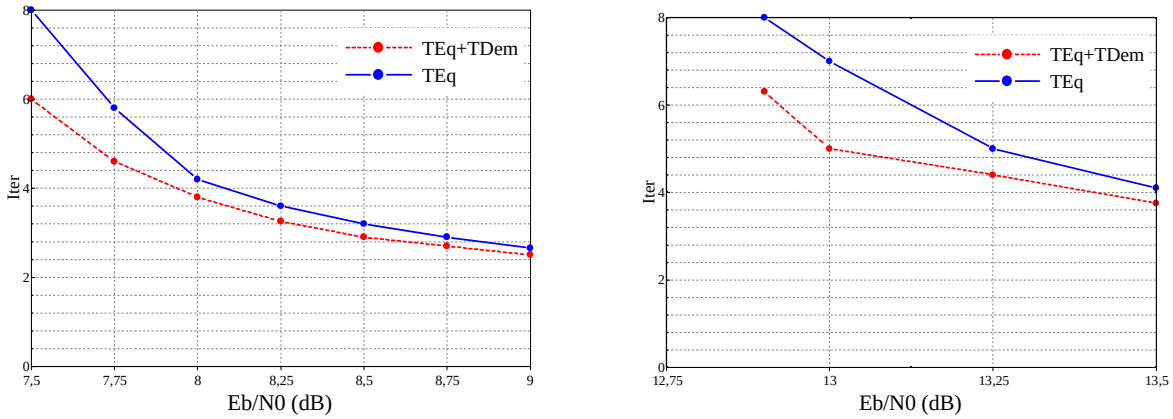
Figure 4.12: BER in function of iterations for 2×2 and 4×4 MIMO SM for QAM64 and QAM256 modulation schemes. E_b/N_0 values are chosen from the E_b/N_0 interval located in the waterfall region. Rayleigh Fast-fading channel, R_c=$\frac{1}{2}$ and N_{CSymb}=768 are considered.

correspondent curves are plotted for a particular E_b/N_0 value as indicated in each subfigure. These E_b/N_0 values are chosen from the E_b/N_0 interval located in the waterfall region. It is shown that for QPSK (*Config. 1* and *Config. 2*) performance simulations that the feedback to the demapper does not improve the BER performance of the MIMO turbo receiver. This is due to the fact that the modulated QPSK symbol is composed of two uncorrelated bits. On the other hand, it is clearly seen from all other configurations that using the *TEq+TDem* mode accelerates the convergence of the iterative MIMO receiver with respect to *TEq*, *i.e.* less iterations are executed to achieve the same BER performance. Taking the example of *Config. 4*, 8 iterations are required to achieve a BER=10^{-5} when *TEq* is applied, while 6 iterations are required for *TEq+TDem* to achieve the same BER. It is worth to note that the reduction in number of iterations depends on the chosen E_b/N_0 value.

Similar behavior is seen for other configurations (different code rates and N_{CSymb} values). The reduction in the number of iterations will lead a priori to improved power consumption, throughput and latency.

It is worth to note that the conducted BER performance analysis in this section can be represented by other form of curves. One of these forms is to target a specific BER and to plot the required number of iterations as function of E_b/N_0. As we can see from Fig. 4.13, to reach the target BER performance

a minimum value of E_b/N_0 is required for each system parameters. For these minimum values, the *TEq* mode requires high number of iterations (e.g. 8 iterations at E_b/N_0=7.5 dB for Fig. 4.13a). Hence, applying the *TEq+TDem* mode will lead to significant less number of iterations (6 iterations for the same example). However, at higher E_b/N_0 values, the *TEq* mode requires few iterations (3 iterations at E_b/N_0=8.75 dB) to achieve the target BER. In this case, the reduction in number of iterations for using the *TEq+TDem* mode will not be significant (no reduction for the same example) since the iterative receiver requires at least a minimum number of iterations to converge. As a result, the *TEq+TDem* mode provides higher gains in iterations number (compared to the *TEq* mode) for the lower E_b/N_0 values achieving the target BER.



(a) QAM16 with 2×2 MIMO SM at BER=10^{-5} **(b)** QAM64 with 4×4 MIMO SM at BER=4.10^{-5}

Figure 4.13: Number of equalization iterations in function of E_b/N_0, to target a specific BER, for QAM16 and QAM64 modulation schemes with different number of antennas. Rayleigh Fast-fading channel, R_c=$\frac{1}{2}$ and N_{CSymb}=768 are considered.

4.5.2 Discussions and Achieved Improvements

In order to evaluate the improvement in complexity when using the receiver mode *TEq+TDem* rather than the receiver mode *TEq*, we define the complexity ratio G_6 as follows.

$$G_6 = \frac{C^{sys}_{TEq} - C^{sys}_{TEq+TDem}}{C^{sys}_{TEq}} \tag{4.31}$$

Converting the number of equalized and modulated symbols into equivalent number of coded symbols (equation (4.27)) and putting equation (4.24) and (4.26) into equation (4.31), G_6 can be written as:

$$G_6 = \frac{c - d}{e} \tag{4.32}$$

where

$$\begin{aligned} c &= (it_{TEq} - It_{TEq+TDem})[\frac{1}{N_r}C^{+}_{eq}(N_r) + \frac{1}{\alpha}C_{dec} + C_{mod}(M)] \\ d &= (it_{TEq+TDem} - 1)C_{dem,TEq+TDem}(M) - (it_{TEq} - 1)C_{dem,TEq}(M) \\ e &= \frac{1}{\alpha}C^{sys}_{TEq} \end{aligned}$$

Modulation	*Config.*	N_r	it	it	G_6			Target
			$TEq+TDem$	TEq	G_6^{Arith}	G_6^R	G_6^W	BER
QPSK	1	4×4	8	8	-	-	-	-
	2	2×2	8	8	-	-	-	-
QAM16	3	4×4	6	8	21.2%	−6.7%	26.9%	10^{-5}
	4	2×2	6	8	7.3%	−9.1%	26.9%	10^{-5}
QAM64	5	4×4	5	7	13.2%	−36.4%	31.1%	4.10^{-5}
	6	2×2	5	7	−12.3%	−39.8%	31.1%	5.10^{-5}
QAM256	7	4×4	4	6	−32.4%	−137%	33.7%	5.10^{-4}
	8	2×2	4	6	−35.2%	−146%	33.7%	4.10^{-4}

(a) Complexity reduction ratios

$C_{TEq}^{sys} - C_{TEq+TDem}^{sys}$		
$Add(1,1)$	Read one-bit	Write one-bit
-	-	-
-	-	-
76028	-209	376
9207	-267	376
42583	-842	391
-7846	-987	391
-49756	-3573	316
-56845	-3875	316

(b) Complexity reduction values

Table 4.7: Complexity reduction in overall number of arithmetic operations, read, and write memory access for using *TEq+TDem* mode rather than *TEq*. $R_c = \frac{1}{2}$.

c designates the reduced complexity for using ***TEq+TDem*** with less iterations ($It_{TEq+TDem}$ instead of It_{TEq}). d designates the added complexity for using ***TEq+TDem*** with more demapping computations ($C_{dem,TEq+TDem}(M)$ instead of $C_{dem,TEq}(M)$) for each iteration. e is equal to C_{TEq}^{sys} divided by α.

This last equation has been used to obtain individually the complexity reductions in terms of arithmetic (G_6^{Arith}), read access (G_6^R), and write (G_6^W) memory access operations as shown in Table 4.7(a).

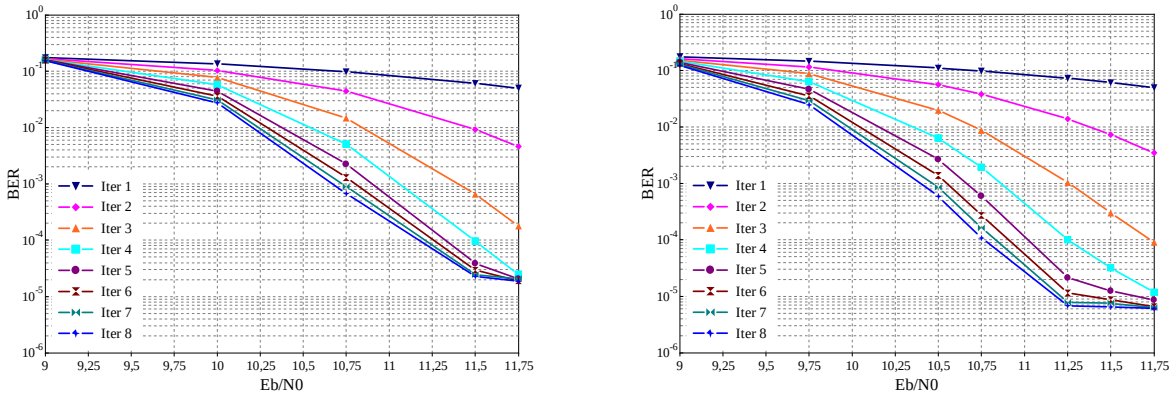
Table 4.7(a) presents the system-level complexity improvements for using ***TEq+TDem*** mode rather than ***TEq*** for the eight configurations. it_{TEq} and the correspondent $it_{TEq+TDem}$ values are taken from Fig. 4.11 and Fig. 4.12 to target the same BER values presented in Table 4.7(a). For *Config. 1* and *Config. 2*, complexity reduction values are not computed since the iterative demapping does not improve the BER performance for QPSK modulation scheme. For *Config. 7*, G_6^{Arith} presents a negative value of -32.4%, which means an increased number of arithmetic operations for using ***TEq+TDem*** comparing to ***TEq***. This result can be extended to QAM256 with 2×2 MIMO SM (-35.2%) since the reduced complexity of the SISO MMSE equalizer is less (c is less) with the same demapping added complexity as for *Config. 7*.

TEq+TDem presents the maximum reduced complexity in terms of arithmetic operations of 21.2% for *Config. 3* since the demapping added complexity corresponding to the constellation size is not significant comparing to the reduced complexity (applying less iterations). Similarly for *Config. 4*, G_6^{Arith} is positive and equals to 7.3%. *Config. 5* shows also a reduced complexity G_6^{Arith} about 13.2%. However for *Config. 6*, G_6^{Arith} is negative since the added complexity to the SISO demapper becomes bigger comparing to the reduced complexity.

Regarding the memory access, Table 4.7(a) shows negative values G_6^R for all the considered configurations which correspond to an increased need of read memory accesses when using ***TEq+TDem***. This increase is due to the conducted search for the closest constellation symbol in the SISO demapper for each iteration. On the other hand, G_6^W shows important reduction values for write memory accesses between 26.9% and 33.7% due to the execution of less turbo decoding processes which require less number of write memory accesses.

In addition to the complexity **ratios** discussed in Table 4.7(a), Table 4.7(b) shows the difference in complexity **values** ($C_{TEq}^{sys} - C_{TEq+TDem}^{sys}$) between ***TEq+TDem*** and ***TEq*** in terms of number of $Add(1,1)$ operations, read and write one-bit memory accesses.

As we see from the Table 4.7(b), the increase in read access memory can be considered small in comparison to the reduced number of $Add(1,1)$ operations and write memory accesses. Taking the example of *Config. 5*, 42583 $Add(1,1)$ operations and 391 one-bit write memory accesses are reduced at the expense of an additional use of 842 one-bit read memory accesses when ***TEq+TDem*** is applied. Thus, the high difference in the magnitude of the reduced arithmetic operations and the



(a) *TEq*: QAM64 with 2×2 MIMO SM **(b)** *TEq+TDem*: QAM64 with 2×2 MIMO SM

Figure 4.14: BER performance simulations for *TEq* and *TEq+TDem* for the transmission of 1536 information bits frame over Rayleigh fast-fading channel without erasure. QAM64 modulation scheme and 2×2 MIMO SM are considered. $R_c = \frac{1}{2}$.

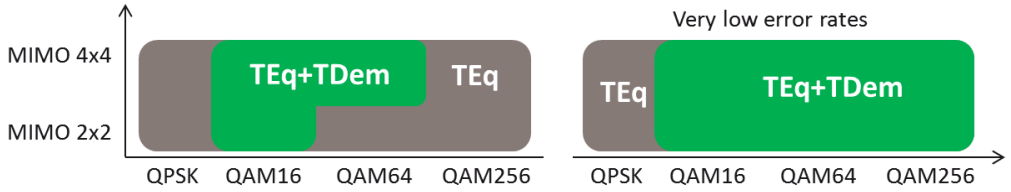


Figure 4.15: Proposal of an adaptive complexity MIMO turbo receiver applying turbo demodulation.

increased read memory accesses gives insights on the convenience of the *TEq+TDem* scheduling to reduce the power consumption of the receiver.

Regarding the error floor region, the *TEq+TDem* mode should be used as it provides more error correction, except for QPSK, as we can see from Fig. 4.14

Fig. 4.15 summarizes the proposed use of the two modes depending on the system parameters.

4.6 Summary

In this chapter an effort is made to present an optimized system-level iterative receiver performing turbo equalization combined with turbo demodulation and turbo decoding.

Convergence speed analysis is crucial in *TEq+TDem* systems in order to tune the number of iterations to be optimal when considering the practical implementation perspectives. Conducted analysis for *TEq+TDem* and *TEq* has demonstrated that omitting one turbo equalization iteration without decreasing the total number of turbo decoding iterations leads to promising complexity reductions while keeping error rate performance almost unaltered. A maximum loss of 0.04 dB is shown for all modulation schemes and code rates in a Rayleigh fast-fading channel with 2×2 and 4×4 MIMO SM. In this regard, the complexity of the receiver was studied taking into account the equivalent arithmetic operations complexity and the memory accesses that should be performed. The number of normalized arithmetic operations can be reduced up to 18.7% in terms of arithmetic operations and up to 16.6% in terms of read access memory for *TEq+TDem*.

The second part of this chapter proposes an adaptive complexity MIMO turbo receiver performing *TEq+TDem*. For QAM16, and for QAM64 with number of transmit and received antennas equals

to 4, additional feedback to the SISO demapper has shown to reduce the overall iterative receiver complexity in terms of computations and write memory accesses along with lower error rate performances. This constitutes a very interesting result as it demonstrates the opposite of what is commonly assumed. In fact, the number of normalized arithmetic operations is reduced, as shown for the considered system configurations, in a range between 7.3% and 21.2% for using *TEq+TDem* rather than *TEq* over Rayleigh fast-fading channel. Similarly, the number of write access memory is reduced in a range between 26.9% and 31.1%. This complexity reduction increases significantly when targeting lower error rates and reduces consequently the power consumption of the iterative MIMO receiver. On the other hand, for QPSK, QAM64 with 2×2 MIMO SM, and QAM256, the *TEq+TDem* receiver should be configured in *TEq* mode which exhibits less complexity for identical error rate performances. Finally, it is worth to note that for very low error rates, except for QPSK, the *TEq+TDem* mode should be used as it provides more error correction.

Conclusions and Perspectives

IN this work, we have studied the convergence speed and the system-level complexity of advanced receivers combining multiple iterative processes. Various communication techniques and system parameters, as specified in emerging wireless communication applications, have been considered. Novel iteration schedulings of inner and outer feedback loops have been proposed to improve the convergence and reduce the overall complexity in terms of arithmetic operations and memory accesses. Furthermore, the conducted analysis and the proposed schedulings demonstrate the effectiveness of the outer feedback loops, even in terms of complexity, compared to the feed forward classical receivers. These results allowed the proposal of original complexity adaptive iterative receivers performing turbo decoding, turbo demodulation and turbo equalization.

Firstly, the basic requirements of advanced wireless digital communication systems have been summarized. Different components of the transmitter such as the channel encoder, the BICM interleaver, the mapper and the MIMO transmission were considered. Low complexity algorithms and parallelism techniques for convolutional decoding, demapping and equalization suitable for hardware implementations have been provided all along this work.

Two ideas for shuffled turbo decoding scheduling, applying a time delay between the processing of the natural and interleaved constituent component decoders, were proposed and analyzed for butterfly and butterfly-replica BCJR metrics computation schemes. The first one has shown a slight improvement in comparison to the classical shuffled decoding scheduling. Meanwhile, the second one leads to a scalable tradeoff between the classical serial and shuffled turbo decoding schedulings in terms complexity and BER performance results.

Furthermore, a convergence speed analysis for turbo demodulation with turbo decoding receiver has been done in order to tune the number of iterations to the exact required ones in order to reduce the overall system complexity. Conducted analysis has demonstrated that omitting two turbo demodulation iterations without decreasing the total number of turbo decoding iterations leads to promising complexity reductions while keeping error rate performance almost unaltered. A maximum loss of 0.15 dB is shown for all modulation schemes and code rates in a fast-fading channel with and without erasure. In this regard, the complexity of the receiver was investigated and normalized taking into account the equivalent arithmetic operations complexity and the memory accesses that should be performed.

Moreover, a complexity adaptive iterative receiver performing TBICM-ID-SSD has been proposed. For low and medium constellation sizes, feedback to the SISO demapper has shown to reduce the complexity in terms of computation and memory access at the receiver side for identical error rate performances. This constitutes a very interesting result as it demonstrates the opposite of what is commonly assumed. On the other had, for high modulation orders, as for QAM64 and QAM256, the TBICM-ID-SSD receiver should be configured in TBICM-SSD mode which provides less complexity for identical error rate performances. It is worth to note that for very low error rates, TBICM-ID-SSD configuration should be used as it provides more error correction in the error floor region.

Based on this study, two further contributions have been proposed as a joint work with two other PhD students. The first one, a joint contribution with Vianney Lapotre, proposes an efficient sizing

of heterogeneous multiprocessor flexible iterative receiver implementing turbo demapping with turbo decoding. In fact, for a given communication requirement many architecture alternatives exist and selecting the right one at design-time and at run-time is an essential issue. The proposed approach defines the mathematical expressions which exhibit the number of heterogeneous cores and their features. Its benefits are illustrated through a flexible multi-processor hardware platform for turbo demodulation with turbo decoding. For the considered case study, platform sizing analysis results demonstrate significant reduction of the area of the iterative receiver. The second contribution, a joint contribution with Oscar Sanchez, proposes to extend the use of the butterfly-replica scheme, originally proposed for shuffled turbo decoding, to full shuffled receiver implementing iterative demapping with turbo decoding. Simulation results show that applying this scheme in the turbo decoder reduces the overall number of iterations by at least one iteration in the waterfall region with respect to the butterfly scheme. In order to evaluate the impact on complexity and throughput, a detailed analysis is provided for different system configurations. When comparing butterfly-replica to butterfly for the same BER performances, the former scheme provides a throughput improvement of around 33%. Significant complexity reductions have been also obtained in terms of arithmetic operations and read access memory for almost all the considered configurations without additional delays or area overhead.

In addition, an effort has been made in this work to propose an optimized system-level iterative receiver performing turbo equalization combined with turbo demodulation and turbo decoding. A convergence speed analysis for *TEq* (without feedback loop to the demapper) and *TEq+TDem* (with feedback loop to the demapper) systems has been done. Conducted analysis has demonstrated that omitting one turbo equalization iteration without decreasing the total number of turbo decoding iterations leads to promising complexity reductions while keeping error rate performance almost unaltered. A maximum loss of 0.04 dB is shown for all modulation schemes and code rates in a fast-fading channel with 2×2 and 4×4 MIMO SM.

These last results allowed the proposal of an adaptive complexity MIMO turbo receiver performing *TEq+TDem*. For QAM16 and for QAM64 with number of transmit and received antennas equals to 4, additional feedback to the SISO demapper has shown to reduce the overall iterative receiver complexity in terms of computations and write memory accesses along with lower error rate performances. This constitutes a very interesting result as it demonstrates the effectiveness of the demapper feedback loop even in terms of complexity.

Perspectives

As perspectives, several research studies can be envisaged:

- Extension of the butterfly vs butterfly-replica analysis to MIMO receivers implementing full shuffled iterative equalization with turbo decoding.
- Extension of the studies presented in this thesis work to other baseband receivers. Additional multi-modes blocks are required to perform functions such as synchronization, OFDM interfacing and channel estimation.
- Analysis of the impact of existing stopping criteria associated with the proposed iteration schedulings and investigation of novel stopping criteria for the combined iterative receivers.
- Integration of the proposed iteration schedulings into the available multi-ASIP hardware prototype.

Résumé en Français

Les normes de communications sans fil, sans cesse en évolution, imposent l'utilisation de techniques modernes telles que les turbocodes, les modulations codées à entrelacement bit (BICM), les constellations d'ordre élevé de modulation d'amplitude en quadrature (QAM), la diversité de constellation (SSD), le multiplexage spatial et codage espace-temps multi-antennes (MIMO) avec des paramètres différents pour des transmissions fiables et de haut débit. L'adoption de ces techniques dans l'émetteur peut influencer l'architecture du récepteur de trois façons : (1) les traitement complexes relatifs aux techniques avancées comme les turbocodes, incite à effectuer un traitement itératif dans le récepteur pour améliorer la performance en termes de taux d'erreurs (2) pour satisfaire l'exigence de haut débit avec un récepteur itératif, le recours au parallélisme est obligatoire et enfin (3) pour assurer le support des différentes techniques et paramètres imposés, des implémentations flexibles, mais aussi de haute performance, sont nécessaires.

En plus de ces exigences techniques liées à la croissance rapide de l'industrie des communications sans fil, l'émergence du traitement itératif et l'étendue de leur principe à toute la chaîne de communication numérique s'imposent actuellement comme la solution algorithmique recherchée pour atteindre les nouvelles performances exigées en termes de qualité de transmission. En témoigne la large adoption des turbocodes et des codes LDPC (*Low-Density Parity-Check*) depuis une quinzaine d'années grâces à la véritable révolution qu'ils apportent en terme de pouvoir de correction d'erreurs de transmission. Un autre exemple est l'adoption de la technique de modulation tournée dans la nouvelle norme de diffusion numérique DVB-T2. Cette adoption, qui est devenue la marque de cette norme, était purement basée sur les gains importants apportés par un processus de turbo-démodulation au niveau du récepteur. Ainsi, la mise en œuvre de systèmes de turbo-communications, communément appelés turbo-récepteurs ou ***récepteurs itératifs***, devient primordiale. Cependant, la généralisation du traitement itératif multiplie la complexité en termes de calculs, d'échanges d'information et de mémorisation et constitue ainsi un défi considérable dans une perspective d'implémentation matérielle.

Problèmes et objectifs de la thèse

En effet, l'ajout au niveau du récepteur, qui intègre déjà un décodage canal itératif (e.g. turbo-décodage), de nouveaux traitements itératifs a montré d'excellentes performances dans des mauvaises conditions de transmission (effacement, multi-trajets, canaux à évanouissement). Toutefois, l'adoption d'un traitement itératif en plus du turbo-décodage est fortement limitée par la complexité supplémentaire engendrée, qui impacte fortement le débit, la latence et la consommation énergétique. Outre l'échange itératif d'informations extrinsèques à l'intérieur du turbo-décodeur, de nouvelles informations extrinsèques sont produites et échangées comme information a priori utilisée par le démappeur et/ou l'égaliseur MIMO. La majeure partie des travaux existants sont au niveau algorithmique et n'ont pas considéré ces techniques d'une perspective d'implémentation matérielle.

Pour faire face à ce problème et permettre une large adoption du traitement itératif, de nouvelles techniques d'optimisation au niveau système doivent être explorées. Ainsi, l'objectif de cette thèse est d'apporter des optimisations au niveau système à travers de nouvelles propositions d'ordonnancement des itérations locales et globales pour le récepteur complet avec turbo-décodage, turbo-égalisation et turbo-démodulation. Il s'agit d'effectuer dans ce cadre une étude approfondie de la rapidité de la convergence du système itératif vis-à-vis de la complexité induite et les différents paramètres, modes de communications et exigences à supporter. Diverses techniques de communication et différents paramètres du système de communication, tels que spécifiés dans les applications émergentes de communication sans fil, doivent être considérés.

Contributions

Pour atteindre les objectifs cités ci-dessus, plusieurs contributions ont été proposées dans le cadre de ce travail de thèse :

1. La première partie de ce travail de thèse s'est concentré sur l'étude et l'analyse de la combinaison des deux traitements itératifs de turbo-démodulation et de turbo-décodage (TBICM-ID-SSD). Les principales contributions accomplies au cours de cette partie peuvent être résumées par la liste suivante :

    - Analyse de la vitesse de convergence de ces deux processus itératifs combinés afin de déterminer le nombre exact d'itérations nécessaires à chaque niveau. Les diagrammes EXIT (*EXtrinsic Information Transfer*) sont utilisés pour une analyse approfondie avec différents ordres de modulation et rendements de codage.
    - Proposition d'un nouveau schéma d'ordonnancement des itérations par l'élimination de deux itérations de démodulation avec une perte raisonnable de performance de moins de 0,15 dB.
    - Analyse et proposition d'une méthode de normalisation de la complexité des opérations arithmétiques et des accès mémoire, qui impactent directement la latence et la consommation énergétique du récepteur itératif. Cette analyse permet d'évaluer les améliorations

dues à l'ordonnancement proposé et les contributions prometteuses en termes de réduction de la complexité du récepteur itératif.

- Étude de la complexité et des performances en termes de taux d'erreurs pour les deux récepteurs itératifs TBICM-ID-SSD et TBICM-SSD (sans retour vers le démodulateur). Cette étude a démontré une réduction considérable de la complexité globale du système en utilisant le mode TBICM-ID-SSD pour les constellations de taille faible ou moyenne (QPSK et QAM16).
- Analyse et proposition d'expressions mathématiques permettant de calculer le minimum nombre de processeurs (décodeurs et démodulateurs SISO) nécessaires pour atteindre un débit donné pour les deux modes TBICM-SSD et TBICM-ID-SSD.
- Étude de la complexité et des performances en termes de taux d'erreurs pour un récepteur TBICM-ID-SSD avec un traitement combiné (shuffled) et pour les schémas de décodage de type "butterfly" et "butterfly-replica". Cette étude a démontré une réduction considérable de la complexité du système en appliquant le schéma "butterfly-replica" pour tous les choix de constellation et pour tous les rendements de codage.

2. La deuxième partie de ce travail de thèse a étendu l'étude ci-dessus pour les récepteurs MIMO combinant turbo-égalisation, turbo-démodulation et turbo-décodage. Les principales contributions apportées dans le cadre de ces travaux peuvent être résumées par la liste suivante :

   - Analyse de la vitesse de convergence de ces trois processus itératifs combinés afin de déterminer le nombre exact d'itérations nécessaires à chaque niveau. Les diagrammes EXIT sont utilisés pour une analyse approfondie avec différents nombres d'antennes, ordres de modulation et rendements de codage.
   - Proposition d'un nouveau schéma d'ordonnancement des itérations par l'élimination d'une seule itération d'égalisation avec une perte raisonnable de performance de moins de 0,04 dB.
   - Analyse de la complexité des opérations arithmétiques et des accès mémoire, qui impactent directement la latence et la consommation énergétique du récepteur itératif. Cette analyse permet d'évaluer les améliorations dues à l'ordonnancement proposé et les contributions prometteuses en termes de réduction de la complexité du récepteur itératif.
   - Proposition d'un récepteur MIMO itératif à complexité adaptative. Ce récepteur a démontré une réduction considérable de la complexité globale du système en appliquant des schémas d'ordonnancement adaptatif des itérations selon la configuration du système.

Structure du manuscrit

Le manuscrit de thèse est organisé en quatre chapitres qui détaillent les contributions citées précédemment.

Chapitre 1

Le premier chapitre introduit les différents paramètres du système de communications qui sont considérés dans cette thèse. Les standards WiMAX, DVB-RCS et DVB-T2 sont brièvement présentés : les paramètres système utilisés pour les simulations lors de ce travail de thèse s'appuient sur ces normes. Ainsi, les techniques considérées et décrites dans ce chapitre sont les suivantes : un codage de canal de type turbocodes double-binaires comme spécifiés dans la norme WiMAX avec plusieurs choix de rendements de codage (1/2 à 5/6), une modulation codée à entrelacement bit avec plusieurs types de modulations (QPSK, QAM16, QAM64, et QAM256), les modulations tournées avec les différentes valeurs d'angles de rotation comme spécifiées dans le standard DVB-T2, et les techniques MIMO avec multiplexage spatial et un nombre d'antennes allant jusqu'à quatre à l'émission et à la réception. Les principes des quatre types de récepteurs itératifs sont ensuite présentés : un récepteur intégrant le seul traitement itératif lié au turbo-décodage, un récepteur combinant turbo-décodage et turbo-démodulation, un récepteur combinant turbo-décodage et turbo-égalisation et un récepteur qui fait interagir les trois fonctions de décodage, de démodulation et d'égalisation.

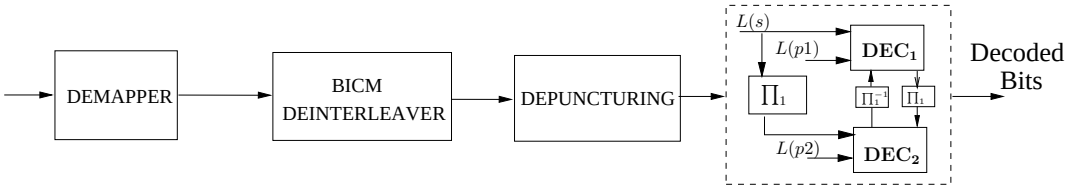


Figure 1: Structure du récepteur avec un traitement itératif au niveau du décodeur de canal : turbo-décodage.

La figure 1 illustre la structure du récepteur avec un traitement itératif au niveau du décodeur de canal. En exploitant la redondance et la diversité ajoutées aux bits d'information au niveau de l'émetteur, le récepteur tente de supprimer les effets de canal afin de récupérer les données d'origine. Chacun des deux composants SISO à entrées et sorties souples du turbo-décodeur traite la trame reçue une seule fois (en prenant en compte les informations a priori reçues de l'autre composant) et transmet ensuite des informations extrinsèques à l'autre composant. Ceci constitue une itération de turbo-décodage.

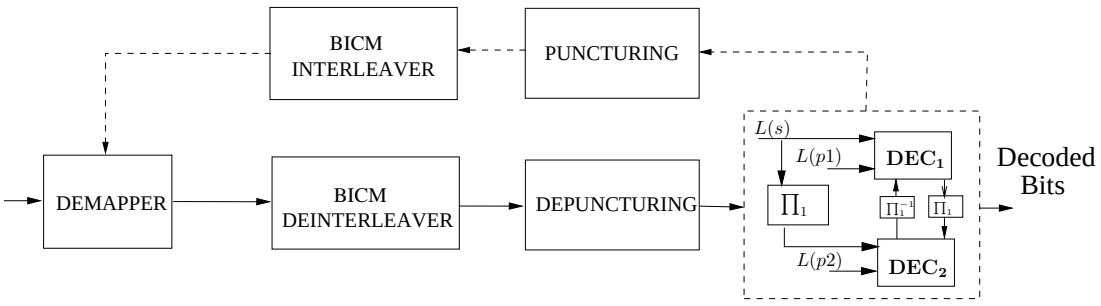


Figure 2: Structure du récepteur itératif combinant turbo-démodulation et turbo-décodage.

L'extension du principe expliqué ci-dessus avec une boucle de retour du turbo-décodeur vers le démappeur SISO peut améliorer les performances en taux d'erreur au prix d'une augmentation de la complexité du récepteur. Une boucle de retour supplémentaire existe ainsi, en plus de la boucle interne au turbo-décodeur, à travers laquelle le turbo-décodeur peut envoyer des informations extrinsèques au démappeur d'une manière itérative. La figure 2 montre la structure du récepteur itératif combinant turbo-démodulation et turbo-décodage. Plusieurs schémas d'ordonnancement des itérations peuvent être trouvés dans l'état de l'art pour ce type de récepteur. A titre d'exemple, les travaux présentés dans [19] utilisent un ordonnancement basé sur l'exécution d'une seule itération de turbo-décodage pour chaque itération de turbo-démodulation.

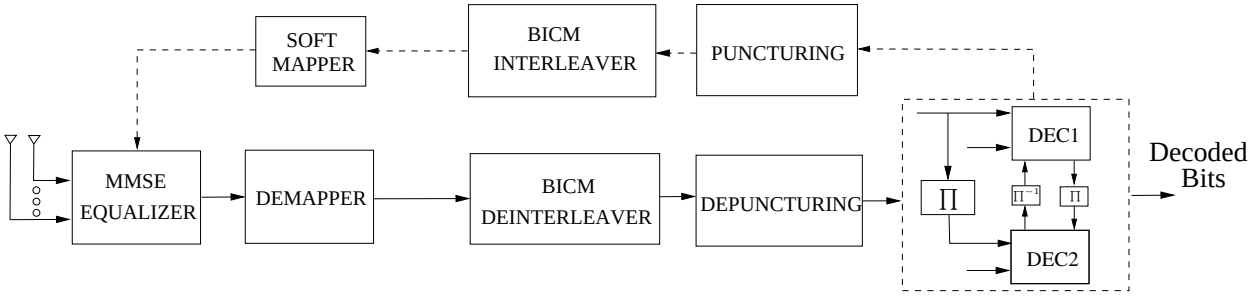


Figure 3: Structure du récepteur itératif combinant turbo-égalisation et turbo-décodage.

De même, pour les systèmes MIMO, une structure de récepteur itératif combinant turbo-égalisation et turbo-décodage est présentée dans la figure 3. Les informations extrinsèques correspondants sont renvoyées à un égaliseur SISO. Plusieurs schémas d'ordonnancement des itérations peuvent être trouvés dans l'état de l'art pour ce type de récepteur. A titre d'exemple, les travaux présentés dans [25, 85] utilisent un ordonnancement basé sur l'exécution d'une seule itération de turbo-décodage pour chaque itération de turbo- égalisation.

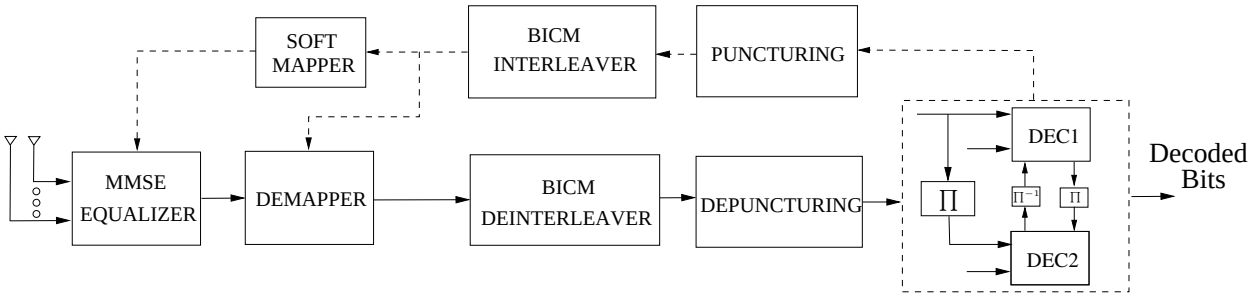


Figure 4: Structure du récepteur itératif combinant turbo-égalisation, turbo-démodulation et turbo-décodage.

Par rapport au récepteur présenté dans le paragraphe précédent, une boucle de retour supplémentaire à partir de turbo-décodeur vers le démappeur peut être appliquée. Ce nouveau récepteur combine ainsi trois traitements itératifs : turbo-égalisation, turbo-démodulation et turbo-décodage (figure 4). A notre connaissance, l'analyse de la vitesse de convergence de ce récepteur n'a pas été adressée dans les travaux existants. Un travail préliminaire peut être trouvé dans [26] où les

auteurs ont présenté un récepteur similaire en utilisant un code convolutif.

Chapitre 2

Le deuxième chapitre est consacré au récepteur avec un seul traitement itératif au niveau du décodage canal : le turbo-décodage des turbocodes convolutifs. Les algorithmes de décodage de codes convolutifs sont rappelés, dont les algorithmes de référence MAP et Max-Log-MAP. Ensuite, les différents niveaux de parallélisme pouvant être mis en œuvre pour accélérer le processus itératif sont décrits. Un premier niveau de parallélisme concerne les calculs des métriques de transition dans le treillis et les calculs des informations extrinsèques. Ce degré de parallélisme dans le calcul des métriques dépend du nombre de transitions du treillis et du schéma de décodage utilisé. Un second niveau de parallélisme consiste à dupliquer les décodeurs SISO eux-mêmes soit pour décoder les données par sous-blocs, soit en dupliquant les décodeurs SISO pour exécuter les deux décodeurs composants du turbo-décodeur en parallèle (traitement combiné ou *shuffled decoding*). La première solution qui consiste à appliquer un décodeur par sous bloc impose certaines contraintes au niveau des caractéristiques des entrelaceurs et une gestion des conditions d'initialisation de chacun des décodeurs SISO. La seconde solution peut induire un accroissement du nombre d'itérations nécessaires pour des degrés de parallélisme élevés. Le dernier niveau de parallélisme consiste à dupliquer le turbo décodeur lui-même mais au prix d'un accroissement considérable de la complexité, en termes notamment de mémoire requise. Nous avons dans ce contexte mené une étude sur l'optimisation du parallélisme lié au décodage combiné (shuffled). Deux techniques de décalage dans le traitement des deux décodeurs composants du turbo-décodeur a été proposée pour permettre de bénéficier plus efficacement des informations a priori échangés. La première technique a montré une légère amélioration par rapport à au mode de décodage combiné (shuffled) classique. La deuxième technique a permis de réaliser un compromis entre le mode de décodage série et le mode de décodage combiné en termes de performances pour chaque itération de turbo-décodage.

Chapitre 3

Le troisième chapitre introduit la fonction de démodulation dans le processus itératif. Un revu de l'état de l'art est d'abord présenté pour positionner les travaux sur cette partie de la thèse. Ensuite les techniques de démodulation SISO à entrées et sorties souples basées sur les algorithmes MAP et Max-Log-MAP sont abordées. Comme pour le turbo-décodage, les différents niveaux de parallélisme du processus de turbo-démodulation sont discutés. La première contribution dans ce cadre qui porte sur l'analyse de la vitesse de convergence des deux processus itératifs combinés (turbo-démodulation et turbo-décodage) afin de déterminer le nombre exact d'itérations nécessaires à chaque niveau est alors présentée. L'utilisation des diagrammes EXIT adaptés à ces récepteurs est tout d'abord décrite. Ensuite, l'intérêt de ce schéma en termes de convergence et de nombre d'itérations est démontré par simulation pour des modulations tournées sur des canaux à évanouissements rapides avec ou sans effacement. L'influence de l'entrelaceur du codage est évaluée, démontrant que celui permettant de

mieux protéger les bits systématiques permet au système de converger en moins d'itérations et avec un tunnel (diagrammes EXIT) de convergence plus favorable.

Différents schémas d'ordonnancement des itérations de turbo-démodulation et de turbo-décodage peuvent être envisagés. Il est ainsi important de mener une étude approfondie pour rechercher le nombre optimal d'itérations pour chacun de ces processus itératifs et le séquencement de ces itérations. L'analyse des résultats de l'étude réalisée a permis de proposer un ordonnancement original où les premières itérations comprennent une boucle de retour vers le démappeur, tandis que les dernières itérations comprennent seulement des itérations de turbo-décodage. L'ordonnancement proposé induit une perte maximale de 0,15 dB pour tous les ordres de modulation et les rendements de codage considérés dans un canal à évanouissement rapide et sans effacements. La réduction de la complexité du récepteur est ensuite évaluée en termes de nombre et de type d'opérations arithmétiques, ainsi que d'accès mémoire en fonction du nombre total d'itérations. Une méthode de normalisation de la complexité a été proposée dans ce contexte. Les résultats de cette analyse ont montré une réduction de la complexité en termes d'opérations arithmétiques normalisées de l'ordre de $15,4\%$ pour la configuration QPSK. Cette réduction augmente significativement pour des modulations d'ordres plus élevés. De même, le nombre d'accès mémoire en lecture diminue de $7,9\%$ pour la configuration QPSK et diminue davantage pour les modulations d'ordres plus élevés.

En outre, plusieurs schémas d'ordonnancement sont comparés pour un même nombre d'opérations quelle que soit l'architecture série ou parallèle. Par conséquent, un récepteur itératif à complexité adaptative appliquant la turbo démodulation avec le turbo décodage est proposé. Pour les constellations de tailles petites et moyennes, le retour vers le démappeur SISO a permis de réduire la complexité du récepteur itératif en termes d'opérations arithmétiques et d'accès mémoires pour des performances identiques en termes de taux d'erreurs. Ceci constitue un résultat très intéressant car il démontre le contraire de ce qui est généralement supposé. En effet, la réduction du nombre d'opérations arithmétiques normalisées est située dans un intervalle compris entre $28,9\%$ et $45,9\%$ pour la configuration QPSK avec l'utilisation du mode TBICM-ID-SSD avec retour vers le démappeur SISO par rapport au mode sans retour TBICM-SSD. Ce dernier applique 6 itérations de turbo-décodage pour un canal à évanouissement de type Rayleigh avec des effacements. De même, le nombre d'accès mémoires en lecture/écriture est réduit dans un intervalle compris entre $29,8\%$ et 47%. D'autre part, pour les constellations d'ordre plus élevé, comme pour QAM64 et QAM256, le récepteur TBICM-ID-SSD doit être configuré en mode de TBICM-SSD qui offre une complexité plus réduite pour des performances identiques en termes de taux d'erreur. Il est important de noter que pour des taux d'erreur très faible, la configuration TBICM-ID-SSD doit être utilisée car elle offre de meilleures performances dans la région d'*error floor*. La figure 5 illustre les modes de fonctionnement proposés en fonction des paramètres du système pour le récepteur itératif à complexité adaptative appliquant turbo-démodulation et turbo-décodage.

Un second type de récepteur a été aussi proposé et étudié dans le contexte de ces travaux. Dans ce récepteur deux itérations de démodulation sont effectuées pour une itération de turbo-décodage, soit un retour vers le SISO démappeur après chacun des 2 décodeurs SISO du turbo-décodeur.

Sur la base de ces études, deux autres contributions ont été proposés comme résultats d'un travail

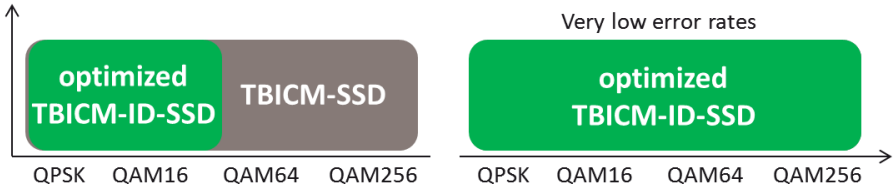


Figure 5: Proposition d'un récepteur itératif à complexité adaptative appliquant turbo-démodulation et turbo-décodage.

en commun avec deux autres doctorants. La première est une contribution commune avec le doctorant Vianney Lapôtre. Elle propose une méthode de dimensionnement efficace d'une plateforme multiprocesseur hétérogène implémentant un récepteur itératif multi-standard combinant turbo-démodulation et turbo-décodage. En fait, pour une application donnée qui impose des exigences spécifiques, de nombreuses alternatives d'architecture existent. Ainsi, sélectionner la bonne architecture au moment de la conception ou en cours d'exécution est primordial pour maximiser l'efficacité de telles plateformes multiprocesseurs. L'approche proposée définit les expressions mathématiques qui permettent de calculer le minimum nombre de processeurs hétérogènes en prenant en considération leurs caractéristiques, les paramètres du système et les besoins applicatifs. Les avantages de l'approche proposée sont illustrés à travers une plateforme matérielle multiprocesseur flexible implémentant un récepteur itératif avec turbo-démodulation et turbo-décodage. Pour l'étude de cas considéré, les résultats d'analyse du dimensionnement proposé montrent une réduction significative de la surface du récepteur itératif. La deuxième contribution est le résultat d'un travail commun avec le doctorant Oscar Sanchez. Il s'agit d'étudier la complexité et les performances en termes de taux d'erreurs pour un récepteur TBICM-ID-SSD avec un traitement combiné (shuffled) et pour les schémas de décodage de type "butterfly" et "butterfly-replica". Les résultats de simulations montrent que l'application du schéma de décodage "butterfly-replica" dans le turbo-décodeur permet de réduire au moins une itération dans la région de convergence par rapport à l'utilisation du schéma "butterfly". Afin d'évaluer l'impact sur la complexité et sur le débit, une analyse détaillée est fournie pour différentes configurations systèmes. Lorsque l'on compare les schémas "butterfly" et "butterfly-replica" pour des performances identiques en taux d'erreurs, le schéma "butterfly-replica" permet une augmentation du débit de l'ordre de 33%. Des réductions significatives de la complexité en termes d'opérations arithmétiques et d'accès mémoire ont été également obtenues pour toutes les configurations considérées sans impacts sur la surface.

Chapitre 4

Le quatrième chapitre étend l'étude menée dans le chapitre 3 sur les systèmes MIMO. Il s'agit d'enrichir la structure du récepteur itératif avec une boucle de retour vers la fonction d'égalisation. L'état de l'art est analysé au début de ce chapitre pour positionner les contributions de ce travail de thèse dans ce domaine. Comme dans le chapitre précédent, l'étude du nombre respectif d'itérations à effectuer pour chacune des fonctions d'égalisation, de démodulation et de décodage est menée. La convergence des différentes configurations est étudiée grâce aux diagrammes EXIT et de nombreux résultats de simulation sont fournis, tout comme une analyse fine de la complexité des différents

schémas d'ordonnancements proposés. Une synthèse des résultats est également réalisée pour permettre de déterminer les conditions d'utilisation des différents récepteurs potentiels avec l'objectif de fournir de bons compromis entre complexité et performance en termes de taux d'erreurs. Les résultats de simulations montrent que le système proposé permettant une réduction de la complexité globale du récepteur en n'effectuant pas le retour vers égaliseur SISO sur toutes les itérations. Une perte maximale de 0,04 dB a été remarqué pour tous les schémas de modulations et les rendements de codage considérés pour un système MIMO de 2x2 et 4x4 antennes avec un multiplexage spatial et un canal à évanouissement rapide sans effacement. La réduction de la complexité est ensuite évaluée en termes de nombre et de type d'opérations arithmétiques, ainsi que d'accès mémoire en fonction du nombre total d'itérations pour les fonctions de base d'égalisation, de démodulation et de turbo-décodage. La réduction de la complexité obtenue atteint $18,7\%$ en termes d'opérations arithmétiques normalisées et $16,6\%$ en termes d'accès mémoire en lecture pour le récepteur itératif proposé combinant turbo-égalisation, turbo-démodulation et turbo-décodage.

Par conséquent, un récepteur itératif MIMO à complexité adaptative a été proposé. Pour les configurations QAM16, et pour QAM64 avec un nombre d'antennes de transmission et de réception égal à 4 : un retour vers le démappeur SISO (récepteur noté *TEq+TDem*) permet de réduire la complexité globale du récepteur itératif pour des performances identiques en termes de taux d'erreurs par rapport au récepteur combinant turbo-égalisation et turbo-décodage (noté *TEq*). Ceci constitue un résultat très intéressant car il démontre le contraire de ce qui est généralement supposé. En effet, la réduction du nombre d'opérations arithmétiques normalisées, pour les configurations considérées dans cette thèse, est située dans un intervalle compris entre $7,3\%$ et $21,2\%$ pour l'utilisation du mode *TEq+TDem* à la place du mode *TEq*. De même, le nombre d'accès mémoire en écriture est réduit dans un intervalle compris entre $26,9\%$ et $31,1\%$. D'autre part, pour les configurations QPSK, QAM64 avec 2 antennes d'émission et de réception, et pour la modulation QAM256, le récepteur MIMO doit être configuré en *TEq*, qui présente dans ces cas une complexité réduite pour des performances identiques en termes de taux d'erreurs. Enfin, il est intéressant de noter que pour des taux d'erreur très faible, sauf pour une modulation QPSK, le mode *TEq+TDem* doit être utilisé car il offre de meilleures performances dans la région d'*error floor*. La figure 6 illustre les modes de fonctionnement proposés en fonction des paramètres du système pour le récepteur itératif à complexité adaptative appliquant turbo-égalisation, turbo-démodulation et turbo-décodage.

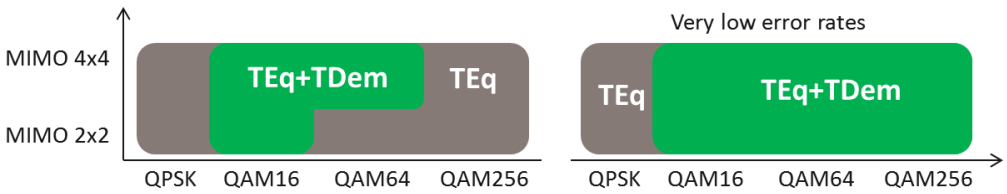


Figure 6: Proposition d'un récepteur MIMO itératif à complexité adaptative appliquant turbo-égalisation, turbo-démodulation et turbo-décodage.

Conclusions et perspectives

Dans ce travail de thèse, nous avons étudié la vitesse de convergence et la complexité au niveau système de récepteurs avancés combinant plusieurs processus itératifs. Diverses techniques de communication et paramètres, tels que spécifiés dans les applications émergentes de communications sans fil, ont été considérés. De nouveaux schémas d'ordonnancement des itérations internes et externes (au turbo-décodeur) ont été proposés pour améliorer la convergence et réduire la complexité globale en termes d'opérations arithmétiques et d'accès mémoire. En outre, l'analyse effectuée et les schémas d'ordonnancement proposés démontrent l'efficacité des boucles de retour externes, même en termes de complexité, par rapport aux récepteurs classiques non itératifs. Ces résultats ont permis la proposition de récepteurs itératifs originaux à complexité adaptative combinant turbo-décodage, turbo-démodulation et turbo-égalisation.

En ce qui concerne les perspectives de travail, plusieurs idées peuvent être étudiées :

- Extension de l'analyse de l'impact des schémas de décodage de type "butterfly" et "butterfly-replica" aux récepteurs MIMO implémentant un traitement combiné (shuffled) complet entre turbo-égalisation et turbo-décodage.
- Extension des études menées dans ce travail de thèse à d'autres fonctions dans la couche physique des récepteurs avancés. D'autres blocs multi-modes sont nécessaires pour exécuter des fonctions telles que la synchronisation, l'accès multiple (e.g. OFDM) et l'estimation de canal.
- Analyse de l'impact des critères d'arrêt existants pour les récepteurs itératifs, associés aux nouveaux schémas d'ordonnancement des itérations proposés, et investigation de nouveaux critères d'arrêt adaptés aux récepteurs combinant plusieurs processus itératifs.
- Intégration des schémas d'ordonnancement des itérations proposés dans la plateforme matérielle multi-ASIP disponible au département Electronique de Télécom Bretagne.

Glossary

3GPP-LTE	3rd Generation Partnership Project-Long Term Evolution
AWGN	Additive White Gaussian Noise
ATM	Asynchronous Transfer Mode
BCJR	Bahl-Cock-Jelinek-Raviv
BICM	Bit-Interleaved Coded Modulation
BPSK	Binary Phase Shift Keying
BTC	Block Turbo Codes
CORDIC	Coordinate Rotation Digital Computer
CC	Convolutional Codes
CTC	Convolutional Turbo Codes
CRSC	Circular Recursive Convolutional Codes
DVB-RCS	Digital Video Broadcasting Return Channel Satellite
DVB-T2	Next Generation Digital Video Broadcasting Terrestrial
DVB-S2	Next Generation Digital Video Broadcasting Over Satellite
EXIT	EXtrinsic Information Transfer
ID	Iterative Demapping
ISI	Inter Symbol Interference
LDPC	Low-Density Parity-Check
LLR	Log Likelihood Ratio
LUT	Look Up Table
MAP	Maximum A Posteriori
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MPEG-2	Motion Picture Experts Group 2
OFDM	Orthogonal Frequency Division Multiplexing
PAM	Pulse Amplitude Modulations

QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RS-CC	Reed-Solomon Convolutional Codes
RSC	Recursive Convolutional Codes
SCCC	Serially Concatenated Convolutional Codes
SD	Sphere Decoding
SGR	Standard Givens Rotation
SISO	Soft In Soft Out
SM	Spatial Multiplexing
SNR	Signal to Noise Ratio
SOVA	Soft Output Viterbi Algorithm
SSD	Signal Space Diversity
STC	Space Time Code
SF	Scaling Factor
TBICM	Turbo Bit-Interleaved Coded Modulation
TTCM	Turbo Trellis Coded Modulation
UMTS	Universal Mobile Telecommunications System
WiMax	Worldwide Interoperability for Microwave Access
WiFi	Wireless Fidelity

Notations

M	Number of bits per modulated symbols
R_c	Turbo encoder code rate
∇	Number of bits per information symbol at the input of the turbo encoder
Φ	Constellation rotation angle in degrees
$\mathcal{X}_{r,l}^{p}$	Symbol set of the constellation
$\mathcal{X}$	Non rotated constellation
$\mathcal{X}_r$	Rotated constellation
N_t	Number of transmit antennas
N_r	Number of receive antennas
u_i	Information bit
d_k	Information symbol
$c_{p,q}$	Coded bit
I	Complex symbol in-phase component
Q	Complex symbol quadrature component
h_q	Rayleigh fast-fading coefficient
ρ_q	Erasure coefficient
P_ρ	Probability of the erasure coefficient
n_q	AWGN complex variable
σ_x^2	Variance of x
$x_{r,q}$	Complex rotated received symbol
$x'_{r,q}$	Complex received symbol after SSD
$s_{r,q}$	Complex rotated transmitted symbol
$s'_{r,q}$	Complex transmitted symbol after SSD
H	Channel complex matrix
Y	MIMO received complex vector
X	MIMO transmitted complex vector

$\hat{X}$	Equalizer *a priori* information complex vector
$\tilde{X}$	Estimated equalizer output complex vector
G	Estimated equalizer real bias vector
W	AWGN complex vector
ν	Number of the component decoder memory elements
α_k	Decoder forward recursion metric
β_k	Decoder backward recursion metric
γ_k	Decoder branch metric
L_{Dec}^{apr}	Turbo decoder *a priori* information
L_{Dec}^{ext}	Turbo decoder extrinsic information
L_{Dec}^{apost}	Turbo decoder *a posteriori* information
A_q	Demapper euclidean distance
$B_{p,q}$	Demapper *a priori* adder
min	Minimum finder
L_{Dem}^{apr}	Demapper *a priori* information
L_{Dem}^{ext}	Demapper extrinsic information
$\mathtt{DEC}_1$, $\mathtt{DEC}_2$	Component decoders
F-B	Forward-Backward scheme
B-R	Butterfly-Replica scheme
B	Butterfly scheme
δ	Normalized delay
Sub	Subtraction operation
Add	Addition operation
$load$	Read access memory operation
$store$	Write access memory operation
$Add(1,1)$	2-input one bit full adder
CASE 1	Re-calculated demapping euclidean distance
CASE 2	Stored demapping euclidean distance
N_{CSymb}	Number of coded symbols per frame
N_{MSymb}	Number of modulated symbols per frame
N_{ESymb}	Number of equalized symbols per frame
TBICM-SSD	Turbo BICM coupled with SSD
TBICM-ID-SSD	Turbo BICM with turbo demodulation coupled with SSD
C_{IDec}	Complexity of the TBICM-SSD scheduling

C_{IDem}	Complexity of the classical TBICM-ID-SSD scheduling
$C_{NEW-IDem}$	Complexity of the proposed TBICM-ID-SSD scheduling
$C_{NEW2-IDem}$	Complexity of the modified-new TBICM-ID-SSD scheduling
C_{dem}^{-}	Demapping complexity without *a priori* computation
C_{dem}^{+}	Demapping complexity with *a priori* computation
C_{dec}	Turbo decoding complexity
C_{eq}^{-}	Equalization complexity without *a priori* computation
C_{eq}^{+}	Equalization complexity with *a priori* computation
TEq	Turbo equalization combined with turbo decoding
TEq+TDem	Turbo equalization combined with turbo demodulation and turbo decoding
C_{TEq}^{sys}	Complexity of the *TEq* scheduling
$C_{TEq+TDem}^{sys}$	Complexity of the *TEq+TDem* scheduling

Bibliography

[1] *802.16 IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, Std., 2004.

[2] R. Gallager, "Low-Density Parity-Check Codes," in *Cambridge, MIT Press*, 1963.

[3] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, may 1998.

[4] C. Shannon, "A mathematical theory of communication," Bell System Technical Journal, Tech. Rep., 1948.

[5] *UMTS Network and Radio Access Technology: Air Interface Techniques for Future Mobile Systems*, Std.

[6] *3GPP Technical Specifications 36.212, Multiplexing and Channel Coding (Release 8)*, Std., 2008.

[7] N. Kiyani and J. Weber, "Iterative demodulation and decoding for rotated MPSK constellations with convolutional coding and signal space diversity," in *Proc. of the IEEE Vehicular Technology Conference (VTC-Fall)*, 2007, pp. 1712–1716.

[8] J. Boutros and E. Viterbo, "Signal space diversity: a power- and bandwidth-efficient diversity technique for the rayleigh fading channel," *IEEE Transactions on Information Theory*, vol. 44, no. 4, pp. 1453–1467, 1998.

[9] *Digital Video Broadcasting (DVB); interaction channel for satellite distribution systems*, Std., 2009.

[10] J. Anderson and S. Hladik, "Tailbiting MAP decoders," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 297–302, feb 1998.

[11] W. Xiang and S. Pietrobon, "A new class of parallel data convolutional codes," in *Proc. of the 6th Australian Workshop on Communications Theory (AusCTW)*, feb. 2005, pp. 84–88.

[12] P. Lee, "Constructions of Rate (n-1)/n Punctured Convolutional Codes with Minimum Required SNR Criterion," *IEEE Transactions on Communications*, vol. 36, no. 10, pp. 1171–1174, 1988.

[13] C. Berrou, M. Jezequel, C. Douillard, and S. Kerouedan, "The Advantages of Non-Binary Turbo Codes," in *Proc. of the IEEE Information Theory Workshop (ITW)*, 2001, pp. 61–63.

[14] C. Douillard, M. Jezequel, C. Berrou, J. Tousch, N. Pham, and N. Brengarth, "The Turbo Code Standard for DVB-RCS," in *Proc. of the International Symposium on Turbo Codes and Related Topics (ISTC)*, 2000, pp. 535–538.

[15] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, oct. 1996.

[16] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873–884, may 1992.

[17] A. Chindapol and J. Ritcey, "Design, analysis, and performance evaluation for BICM-ID with square QAM constellations in Rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 944–957, 2001.

[18] J. Tan and G. Stuber, "Analysis and design of symbol mappers for iteratively decoded BICM," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 662–672, march 2005.

[19] C. Abdel Nour and C. Douillard, "On lowering the error floor of high order turbo BICM schemes over fading channels," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, nov. 2006, pp. 1–5.

[20] ——, "Improving BICM performance of QAM constellations for broadcasting applications," in *Proc. of the International Symposium on Turbo Codes and Related Topics (ISTC)*, 2008, pp. 55–60.

[21] L. Meng, C. Nour, C. Jego, and C. Douillard, "Design of rotated QAM mapper/demapper for the DVB-T2 standard," in *Proc. of the IEEE Workshop on Signal Processing Systems (SiPS)*, oct. 2009, pp. 018–023.

[22] S. A. Barbulescu, W. Farrell, P. Gray, and M. Rice, "Bandwidth efficient turbo coding for high speed mobile satellite communications," in *Proc. of the International Symposium on Turbo Codes and Related Topics (ISTC)*, 1997, pp. 119–126.

[23] S. L. Goff, A. Glavieux, and C. Berrou, "Turbo-codes and high spectral efficiency modulation," in *Proc. of the IEEE International Conference on Communications (ICC)*, vol. 2, 1994, pp. 645–649.

[24] C. Douillard, M. Jézéquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.

[25] A. Jafri, A. Baghdadi, and M. Jézéquel, "Parallel MIMO Turbo Equalization," *IEEE Communications Letters*, vol. 15, no. 3, pp. 290–292, march 2011.

[26] K. Amis, G. Sicot, and D. Leroux, "Reduced complexity near-optimal iterative receiver for WiMax full-rate space time code," in *Proc. of the International Symposium on Turbo Codes and related topics (ISTC)*, 2008.

[27] J. Forney, G., R. Gallager, G. Lang, F. Longstaff, and S. Qureshi, "Efficient modulation for band-limited channels," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 5, pp. 632–647, sept. 1984.

[28] I. Abramovici and S. Shamai, "On turbo encoded BICM," *Annales des telecommunications*, vol. 54, pp. 225–234, 1999.

[29] P. Robertson and T. Worz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 206–218, feb. 1998.

[30] R. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Transactions on Information Theory*, vol. 9, no. 2, pp. 64–74, 1963.

[31] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[32] J. Forney, G.D., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[33] J. Hagenauer and P. Hoeher, "A viterbi algorithm with soft-decision outputs and its applications," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 3, 1989, pp. 1680–1686.

[34] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[35] S. B., *Digital Communications: Fundamentals and Applications. Second edition Fundamentals of Turbo Codes*. Prentice Hall, 2001.

[36] O. Muller, A. Baghdadi, and M. Jezequel, "Exploring parallel processing levels for convolutional turbo decoding," in *Proc. of the IEEE International Conference on Information & Communication Technologies: From Theory To Applications (ICTTA)*, vol. 2, april 2006, pp. 2353–2358.

[37] ——, "From Parallelism Levels to a Multi-ASIP Architecture for Turbo Decoding," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 1, pp. 92–102, jan. 2009.

[38] G. Masera, G. Piccinini, M. Roch, and M. Zamboni, "VLSI architectures for turbo codes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 3, pp. 369–379, sept. 1999.

[39] E. Boutillon, W. Gross, and P. Gulak, "VLSI architectures for the MAP algorithm," *IEEE Transactions on Communications*, vol. 51, no. 2, pp. 175–185, feb. 2003.

[40] Y. Zhang and K. Parhi, "Parallel turbo decoding," in *Proc. of the International Symposium on Circuits and Systems (ISCAS)*, vol. 2, no. 2, may 2004, pp. 509–512.

[41] H. Moussa, "Architecture de Réseaux sur Puce Pour Décodeur Canal Multiprocesseurs," Ph.D. dissertation, ELEC - Dépt. Electronique, Telecom Bretagne, 2009.

[42] O. Muller, "Architectures multiprocesseurs monopuces génériques pour turbo-communications haut-débit," Ph.D. dissertation, Université de Bretagne-Sud, 2007.

[43] J. Zhang and M. Fossorier, "Shuffled iterative decoding," *IEEE Transactions on Communications*, vol. 53, no. 2, pp. 209–213, feb. 2005.

[44] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollarab, "A Soft-Input Soft-Output Maximum A Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes," *TDA Progress Report*, no. 42-127, nov. 1996.

[45] D. Divsalar and F. Pollara, "Multiple turbo codes for deep space communications," *JPL TDA Prog. Rep.*, pp. 71–78, may 1995.

[46] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," in *Proc. of the IEEE International Symposium on Information Theory (ISIT)*, 1997, p. 96.

[47] X. Li and J. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Communications Letters*, vol. 1, no. 6, pp. 169–171, 1997.

[48] A. R. Jafri, "Architectures multi-ASIP pour turbo récepteur flexible," Ph.D. dissertation, Electronics Dept., Telecom Bretagne, 2011.

[49] A. R. Jafri, A. Baghdadi, and M. Jezequel, "Exploring parallel processing levels in turbo demodulation," in *Proc. of the International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, sept. 2010, pp. 359–363.

[50] A. Jafri, A. Baghdadi, and M. Jezequel, "ASIP-Based Universal Demapper for Multiwireless Standards," *IEEE Embedded Systems Letters*, vol. 1, no. 1, pp. 9–13, may 2009.

[51] A. Jafri, A. Baghdadi, and M. Jézéquel, "Rapid design and prototyping of universal soft demapper," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, June 2010, pp. 3769–3772.

[52] T. Clevorn, F. Oldewurtel, S. Godtmann, and P. Vary, "Iterative demodulation for DVB-S2," in *Proc. of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, vol. 4, sept. 2005, pp. 2576–2580.

[53] S. Ten Brink, J. Speidel, and R.-H. Yan, "Iterative demapping and decoding for multilevel modulation," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 1, 1998, pp. 579–584.

[54] S. T. Brink, "Convergence of iterative decoding," *IEEE Electronics Letters*, vol. 35, no. 13, pp. 1117–1119, 1999.

[55] S. Ten Brink, "Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes," *IEEE Transactions on Communications,*, vol. 49, no. 10, pp. 1727 –1737, Oct 2001.

[56] F. Brannstrom, L. Rasmussen, and A. Grant, "Convergence Analysis and Optimal Scheduling for Multiple Concatenated Codes," *IEEE Transactions on Information Theory,*, vol. 51, no. 9, pp. 3354 – 3364, Sept. 2005.

[57] J. Hagenauer, "The EXIT chart - introduction to extrinsic information transfer," in *Proc. of the 12th European Signal Processing Conference (EUSIPCO)*, 2004, pp. 1541–1548.

[58] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," The Telecommunications and Data Acquisition Report, Tech. Rep. 56-65, 1995.

[59] M. Woh, S. Seo, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, "AnySP: Anytime Anywhere Anyway Signal Processing," *IEEE Micro*, vol. 30, no. 1, pp. 81–91, 2010.

[60] J. Glossner, D. Iancu, M. Moudgill, G. Nacer, S. Jinturkar, S. Stanley, and M. Schulte, "The sandbridge SB3011 platform," *EURASIP Journal on Embedded Systems*, pp. 16–16, 2007.

[61] C. Jalier, D. Lattard, G. Sassatelli, P. Benoit, and L. Torres, "Flexible and distributed real-time control on a 4G telecom MPSoC," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 3961–3964.

[62] F. Clermidy, C. Bernard, R. Lemaire, J. Martin, I. Miro-Panades, Y. Thonnart, P. Vivet, and N. Wehn, "MAGALI: A Network-on-Chip based multi-core system-on-chip for MIMO 4G SDR," in *Proc. of the IEEE International Conference on IC Design and Technology (ICICDT)*, 2010, pp. 74–77.

[63] U. Ramacher, "Software-Defined Radio Prospects for Multistandard Mobile Phones," *Computer*, vol. 40, no. 10, pp. 62–69, 2007.

[64] T. Limberg, M. Winter, M. Bimberg, R. Klemm, E. Matus, M. Tavares, G. Fettweis, H. Ahlendorf, and P. Robelly, "A fully programmable 40 GOPS SDR single chip baseband for LTE/WiMAX terminals," in *Proc. of the Solid-State Circuits Conference (ESSCIRC)*, sept. 2008, pp. 466–469.

[65] J. Declerck, P. Raghavan, F. Naessens, T. Aa, L. Hollevoet, A. Dejonghe, and L. Van der Perre, "SDR platform for 802.11n and 3-GPP LTE," in *Proc. of the International Conference on Embedded Computer Systems (SAMOS)*, 2010, pp. 318–323.

[66] T. Suzuki, H. Yamada, T. Yamagishi, D. Takeda, K. Horisaki, T. Vander Aa, T. Fujisawa, L. Perre, and Y. Unekawa, "High-Throughput, Low-Power Software-Defined Radio Using Reconfigurable Processors," *IEEE Micro*, vol. 31, no. 6, pp. 19–28, 2011.

[67] C. Brehm, T. Ilnseher, and N. Wehn, "A scalable multi-ASIP architecture for standard compliant trellis decoding," in *Proc. of the International SoC Design Conference (ISOCC)*, 2011, pp. 349–352.

[68] T. Vogt, C. Neeb, and N. Wehn, "A reconfigurable multi-processor platform for convolutional and turbo decoding," in *Proc. of the International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, 2006, pp. 16–23.

[69] P. Murugappa, A.-K. R., A. Baghdadi, and M. Jézéquel, "A Flexible High Throughput Multi-ASIP Architecture for LDPC and Turbo Decoding," in *Proc. of the Design, Automation and Test in Europe Conference & Exhibition (DATE)*, 2011.

[70] A. R. Jafri, A. Baghdadi, and M. Jezequel, "FPGA Prototype of Flexible Heterogeneous multi-ASIP NoC-based Unified Turbo Receiver," in *University Booth of the Design, Automation and Test in Europe Conference & Exhibition (DATE)*, 2011.

[71] ——, "ASIP-Based Universal Demapper for Multiwireless Standards," *IEEE Embedded Systems Letters*, vol. 1, no. 1, pp. 9–13, 2009.

[72] O. Muller, A. Baghdadi, and M. Jezequel, "Parallelism Efficiency in Convolutional Turbo Decoding," *EURASIP Journal on Advances in Signal Processing*, 2010.

[73] R. Al-Khayat, P. Murugappa, A. Baghdadi, and M. Jezequel, "Area and throughput optimized ASIP for multi-standard turbo decoding," in *Proc. of IEEE International Symposium on the Rapid System Prototyping (RSP)*, may 2011, pp. 79–848.

[74] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, 2005.

[75] G. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 18, no. 3, pp. 363–378, 1972.

[76] U. Finkce and M. Pohst, "Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis," *Math. Comput.*, vol. 44, pp. 463–471, apr. 1985.

[77] O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Communications Letters*, vol. 4, no. 5, pp. 161–163, 2000.

[78] M. Witzke, "Linear and widely linear filtering applied to iterative detection of generalized MIMO signals," *Annales des Télécommunications*, vol. 60, no. 1-2, pp. 147–168, 2005.

[79] M. Karkooti, J. R. Cavallaro, and C. Dick, "FPGA Implementation of Matrix Inversion Using QRD-RLS Algorithm," in *Proc. of the Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, oct. 2005, pp. 1625–1629.

[80] C. Laot, R. L. Bidan, and D. Leroux, "Low Complexity MMSE Turbo Equalization: A Possible Solution for EDGE," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 965–974, may 2005.

[81] D. Karakolah, "Conception et prototypage d'un récepteur itératif pour des systèmes de transmision MIMO avec précodage linéaire," Ph.D. dissertation, Electronics Dept., Telecom Bretagne, 2009.

[82] D. Karakolah, C. Jego, C. Langlais, and M. Jezequel, "Design of an iterative receiver for linearly precoded MIMO systems," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, may 2009, pp. 597–600.

[83] ——, "Architecture dedicated to the MMSE equalizer of iterative receiver for linearly precoded MIMO systems," in *Proc. of the IEEE International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA)*, april 2008, pp. 1–6.

[84] C. Studer, S. Fateh, and D. Seethaler, "ASIC Implementation of Soft-Input Soft-Output MIMO Detection Using MMSE Parallel Interference Cancellation," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, july 2011.

[85] L. Boher, "Etude et mise en oeuvre de récepteurs itératifs pour systèmes MIMO," Ph.D. dissertation, INSA Rennes - Institut National des Sciences Appliquées de Rennes, Laboratoire Broadband Wireless Access, France Telecom division R&D, 2008.

[86] L. Boher, M. Helard, and R. Rabineau, "Turbo-Coded MIMO Iterative Receiver with Bit Per Bit Interference Cancellation for M-QAM Gray Mapping Modulation," in *Proc. of the IEEE Vehicular Technology Conference, 2007. VTC-Spring*, april 2007, pp. 2394–2398.

[87] ——, "MIMO Iterative Receiver with Bit Per Bit Interference Cancellation," in *Proc. of the International Symposium on Wireless Communication Systems (ISWCS)*, sept. 2006, pp. 804–808.

[88] L. Boher, R. Rabineau, and M. Helard, "An Efficient MMSE Equalizer Implementation for 4x4 MIMO-OFDM Systems in Frequency Selective Fast Varying Channels," in *Proc. of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRCC)*, sept. 2007, pp. 1 –5.

[89] ——, "FPGA Implementation of an Iterative Receiver for MIMO-OFDM Systems," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 6, pp. 857–866, august 2008.

[90] A. Raza Jafri, A. Baghdadi, and M. Jezequel, "Parallel MIMO Turbo Equalization," *IEEE Communications Letters*, vol. 15, no. 3, pp. 290–292, 2011.

[91] A. Jafri, D. Karakolah, A. Baghdadi, and M. Jezequel, "ASIP-based flexible MMSE-IC Linear Equalizer for MIMO turbo-equalization applications," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, april 2009, pp. 1620–1625.

[92] A. Jafri, A. Baghdadi, and M. Jezequel, "Rapid Prototyping of ASIP-based Flexible MMSE-IC Linear Equalizer," in *Proc. of the IEEE International Symposium on Rapid System Prototyping (RSP)*, june 2009, pp. 130–133.

[93] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, October 1998.

[94] J. C. Belfiore, G. Rekaya, and E. Viterbo, "The golden code: a 2×2 full-rate space-time code with nonvanishing determinants," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1432–1436, April 2005.

[95] M. T. Gamba, G. Masera, and A. Baghdadi, "Iterative MIMO Detection: Flexibility and Convergence Analysis of SISO List Sphere Decoding and Linear MMSE Detection," in *Proc. of the International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, sept. 2010.

[96] F. Borlenghi, E. Witte, G. Ascheid, H. Meyr, and A. Burg, "A 772Mbit/s 8.81bit/nJ 90nm CMOS soft-input soft-output sphere decoder," in *Proc. of the IEEE Asian Solid State Circuits Conference (A-SSCC)*, nov. 2011, pp. 297–300.

[97] F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg, "A 2.78 mm2 65 nm CMOS gigabit MIMO iterative detection and decoding receiver," in *Proc. of the IEEE Solid-State Circuits Conference (ESSCIRC)*, sept. 2012, pp. 65–68.

[98] D. Zhang, I.-W. Lai, K. Nikitopoulos, and G. Ascheid, "Informed message update for iterative MIMO demapping and turbo decoding," in *Proc. of the IEEE International Symposium on Information Theory and its Applications (ISITA)*, oct. 2010, pp. 873–878.

[99] K. Nikitopoulos and G. Ascheid, "Approximate MIMO Iterative Processing With Adjustable Complexity Requirements," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 2, pp. 639–650, feb. 2012.

[100] M. Witzke, S. Baro, F. Schreckenbach, and J. Hagenauer, "Iterative Detection of MIMO Signals with Linear Detectors," in *Proc. of the Asilomar Conference on Signals, Systems and Computers*, vol. 1, nov. 2002, pp. 289–293.

[101] M. Tuchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 673–683, march 2002.

[102] R. L. Bidan, "Turbo-equalization for bandwidth-efficient digital communications over frequency-selective channels," Ph.D. dissertation, INSA de Rennes, Rennes, France, 2003.

[103] J. Le Masson, "Systèmes de transmission avec précodage linéaire et traitement itératif," Ph.D. dissertation, ELEC - Dépt. Electronique, Telecom Bretagne, 2005.

[104] G. H. Golub and C. F. Van Loan, *Matrix Computations, 3rd Edition*. JHU Press, 1996.

List of publications

Journals

[1] S. Haddad, A. Baghdadi, and M. Jezequel, "On the Convergence Speed of Turbo Demodulation with Turbo Decoding", *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4452-4458, 2012.

[2] S. Haddad, A. Baghdadi, and M. Jezequel, "Complexity Adaptive Iterative Receiver Performing TBCIM-ID-SSD", *EURASIP Journal on Advances in Signal Processing*, 2012.

[3] S. Haddad, A. Baghdadi, and M. Jezequel, "On the Convergence Speed of MIMO Turbo Receiver", *IEEE Communication Letters*, submitted, under revision, 2012.

Conferences

[4] V. Lapotre, S. Haddad, A. Baghdadi, and M. Jezequel, "An analytical approach for sizing of heterogeneous multiprocessor flexible platform for iterative demapping and channel decoding", *Accepted in the International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Cancun, Mexico, 2012.

[5] S. Haddad, A. Baghdadi, and M. Jezequel, "Adaptive Complexity MIMO Turbo Receiver Applying Turbo Demodulation", *Accepted in the 7th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Gothenburg, Sweden, 2012.

[6] S. Haddad, O. Sanchez, A. Baghdadi, and M. Jezequel, "Complexity Reduction of Shuffled Parallel Iterative Demodulation with Turbo Decoding", *In Proc. of the International Conference on Telecommunications (ICT)*, Jounieh, Lebanon, 2012.

[7] S. Haddad, A. Baghdadi, and M. Jezequel, "Reducing the Number of Iterations in Iterative Demodulation with Turbo Decoding", *In Proc. of the International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 2011.

[8] C. Langlais, S. Haddad, Y. Louet, N. Mazouz, "Clipping Noise Mitigation with Capacity Approaching FEC Codes for PAPR Reduction of OFDM Signals", *In Proc. of the International Workshop on Multi-Carrier Systems & Solutions (MC-SS)*, Herrshing, Germany, 2011.

[9] S. Haddad, A. Baghdadi, and M. Jezequel, "Convergence and Complexity Analysis of Turbo Demodulation with Turbo Decoding", *In GDR SoC-SiP: Groupe de recherche System on Chip - System in Package, Colloque National*, Paris, France, 13-15 June 2012.

[10] V. Lapotre, G. Gogniat, A. Baghdadi, S. Haddad, J. P. Diguet, J. Shield, “Management of Reconfigurable Multi-Standards ASIP-based Receiver”, *In GDR SoC-SiP: Groupe de recherche System on Chip - System in Package, Colloque National*, Lyon, France, 15-17 June 2011.